M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
                                                                    G  1
**FILE**ID**BASRSTSFI






BBBBBBBB      AAAAAA      SSSSSSSS  RRRRRRRR      SSSSSSSS  TTTTTTTTTT    SSSSSSSS  FFFFFFFFFF    IIIIII
BBBBBBBB      AAAAAA      SSSSSSSS  RRRRRRRR      SSSSSSSS  TTTTTTTTTT    SSSSSSSS  FFFFFFFFFF    IIIIII
BB      BB  AA      AA  SS          RR      RR  SS              TT      SS          FF              II
BB      BB  AA      AA  SS          RR      RR  SS              TT      SS          FF              II
BB      BB  AA      AA  SS          RR      RR  SS              TT      SS          FF              II
BB      BB  AA      AA  SS          RR      RR  SS              TT      SS          FF              II
BBBBBBBB    AA      AA    SSSSSS    RRRRRRRR      SSSSSS        TT        SSSSSS    FFFFFFFF        II
BBBBBBBB    AA      AA    SSSSSS    RRRRRRRR      SSSSSS        TT        SSSSSS    FFFFFFFF        II
BB      BB  AAAAAAAAAA          SS  RR  RR              SS      TT              SS  FF              II
BB      BB  AAAAAAAAAA          SS  RR  RR              SS      TT              SS  FF              II
BB      BB  AA      AA          SS  RR    RR            SS      TT              SS  FF              II          ....
BB      BB  AA      AA          SS  RR    RR            SS      TT              SS  FF              II          ....
BBBBBBBB    AA      AA  SSSSSSSS    RR      RR  SSSSSSSS        TT      SSSSSSSS    FF            IIIIII        ....
BBBBBBBB    AA      AA  SSSSSSSS    RR      RR  SSSSSSSS        TT      SSSSSSSS    FF            IIIIII        ....


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0001 0 MODULE BAS$RSTS_FIELD (                             ! FIELD statement
0002 0                 IDENT = '1-023'                    ! File: BASRSTSFI.B32 Edit: MDL1023
0003 0                 ) =
0004 1 BEGIN
0005 1 !
0006 1 !****************************************************************************
0007 1 !*                                                                          *
0008 1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0009 1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0010 1 !*  ALL RIGHTS RESERVED.                                                    *
0011 1 !*                                                                          *
0012 1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0013 1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0014 1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0015 1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0016 1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0017 1 !*  TRANSFERRED.                                                            *
0018 1 !*                                                                          *
0019 1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0020 1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0021 1 !*  CORPORATION.                                                            *
0022 1 !*                                                                          *
0023 1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0024 1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0025 1 !*                                                                          *
0026 1 !*                                                                          *
0027 1 !****************************************************************************
0028 1 !
0029 1
0030 1 !++
0031 1 ! FACILITY:  VAX-11 BASIC Miscellaneous
0032 1 !
0033 1 ! ABSTRACT:
0034 1 !
0035 1 !       This module contains the RSTS-compatable FIELD functions.
0036 1 !       A FIELD variable is semi-interpreted, and routines in this
0037 1 !       module "declare" such a variable, copy data to and from it,
0038 1 !       and purge it when the block it was in is exited.
0039 1 !
0040 1 ! ENVIRONMENT:  VAX-11 User Mode
0041 1 !
0042 1 ! AUTHOR: John Sauter, CREATION DATE: 27-FEB-1979
0043 1 !
0044 1 ! MODIFIED BY:
0045 1 !
0046 1 ! 1-001 - Original.  JBS 27-FEB-1979
0047 1 ! 1-002 - Rearrange FIELD_SET so that the compiler can call it conveniently
0048 1 !          once for a FIELD statement.  JBS 01-MAR-1979
0049 1 ! 1-003 - Add a statement type parameter to FIELD_COPY.  JBS 02-APR-1979
0050 1 ! 1-004 - Correct STR$COPY to STR$COPY_DX.  JBS 03-APR-1979
0051 1 ! 1-005 - Re-order some parameters to make things easier on the BASIC-PLUS-2
0052 1 !          compiler.  JBS 18-MAY-1979
0053 1 ! 1-006 - Today the compiler began producing code for the FIELD
0054 1 !          statement, so begin debugging.  JBS 22-MAY-1979
0055 1 ! 1-007 - Add OPEN, CLOSE and KILL entry points.  JBS 24-MAY-1979
0056 1 ! 1-008 - Complete coding of the new entry points.  JBS 25-MAY-1979
0057 1 ! 1-009 - Add BAS$$FIELD_INIT.  JBS 04-JUN-1979
```

```
;    58          0058   1 ! 1-010 - Fix a bug in KILL which made it run forever.  JBS 07-JUN-1979
;    59          0059   1 ! 1-011 - Allow a virtual array to be fielded, but only if it is used
;    60          0060   1 !           exclusively for block I/O.  JBS 22-JUN-1979
;    61          0061   1 ! 1-012 - Add BAS$FIELD_COP_R.  JBS 13-JUL-1979
;    62          0062   1 ! 1-013 - Change calls to STR$COPY.  JBS 16-JUL-1979
;    63          0063   1 ! 1-014 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
;    64          0064   1 ! 1-015 - Call STR$FREE1_DX only for variables not already FIELDed.
;    65          0065   1 !           JBS 02-AUG-1979
;    66          0066   1 ! 1-016 - Call BAS$$CB_GET, so we don't have to be in the sharable
;    67          0067   1 !           library.  JBS 22-AUG-1979
;    68          0068   1 ! 1-017 - Make negative string lengths back up the address but set
;    69          0069   1 !           the variable's length to zero.  JBS 28-FEB-1980
;    70          0070   1 ! 1-018 - When opening a file, validate variables which may have been
;    71          0071   1 !           invalidated by the implied close.  JBS 23-MAY-1980
;    72          0072   1 ! 1-019 - Add 2 to lub$k_ilun_min before calling bas$$cb_push to force an
;    73          0073   1 !         error when -7 or -8 LUNs are used, at the same time put the code that
;    74          0074   1 !         opens channel 0 for upgrading bas$field_set to the level of the other
;    75          0075   1 !         I/O support routines.  FM 17-sep-80
;    76          0076   1 ! 1-020 - Make SYM$Q_ROOT global so that BAS$CLOSE can access it.
;    77          0077   1 !         PL 2-JUN-81
;    78          0078   1 ! 1-021 - BAS$CLOSE will not need SYM$Q_ROOT after all, so make it OWN again.
;    79          0079   1 !         PL 16-Jun-81
;    80          0080   1 ! 1-022 - Undo 19.  We can now do I/O to #0, becuase BAS$PUT will use foreign
;    81          0081   1 !         buffer mechanism to do PUTs to #0.  FM 9-JUL-81.
;    82          0082   1 ! 1-023 - set LUB$V_FIELD_USE when a FIELD statement is executed, so that
;    83          0083   1 !         BAS$CLOSE can tell if there is FIELDing on the channel.  Clear it
;    84          0084   1 !         when the channel is closed.  MDL 29-Mar-1984
;    85          0085   1 !--
;    86          0086   1
;    87          0087   1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1
1 REQUIRE 'RTLIN:OTSLNK';                                 ! Define linkages
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$FIELD_SET : NOVALUE,                            ! Process a FIELD statement
1     BAS$FIELD_VAR : CALL_CCB NOVALUE,                   ! Declare a FIELD variable
1     BAS$FIELD_CLEAR : NOVALUE,                          ! Undeclare a FIELD variable
1     BAS$FIELD_COPY : NOVALUE,                           ! Reference such a variable
1     BAS$FIELD_COP_R : NOVALUE,                          ! Reference such a variable
1     BAS$FIELD_PURGE : NOVALUE,                          ! Purge field variables
1     BAS$FIELD_OPEN : NOVALUE,                           ! A file was just opened
1     BAS$FIELD_CLOSE : NOVALUE,                          ! A file was just closed
1     BAS$$FIELD_KILL : CALL_CCB NOVALUE,                 ! CLOSE appendage
1     BAS$$FIELD_INIT : NOVALUE;                          ! Initialize the FIELD list
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'RTLIN:RTLPSECT';                               ! Macros for defining psects
1
1 REQUIRE 'RTLML:OTSLUB';                                 ! Logical unit block definitions
1
1 REQUIRE 'RTLML:OTSISB';                                 ! ISB definitions
1
1 REQUIRE 'RTLIN:BASFRAME';                               ! BASIC frame structure
1
1 LIBRARY 'RTLSTARLE';                                    ! System definitions
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1
1 LITERAL
1     STMT_TYPE_LSET = 0,                                 ! LSET statement
1     STMT_TYPE_RSET = 1;                                 ! RSET statement
1
1 !
1 ! PSECTS:
1 !
```

```
DECLARE_PSECTS (BAS);                                   ! Declare psects for BAS$ facility
!
! OWN STORAGE:
!

OWN
    SYM$Q_ROOT : VECTOR [2] INITIAL (0, 0);             ! Root for symbol table

!
! EXTERNAL REFERENCES:
!

EXTERNAL ROUTINE
    BAS$$STOP : NOVALUE,                                ! Signal fatal error
    BAS$$STOP_IO : NOVALUE,                             ! Signal fatal I/O error
    LIB$GET_VM,                                         ! Get virtual memory
    LIB$FREE_VM,                                        ! Free virtual memory
    BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,                 ! Load register CCB
    BAS$$CB_POP : JSB_CB_POP NOVALUE,                   ! Done with register CCB
    BAS$$CB_GET : JSB_CB_GET NOVALUE,                   ! Load CCB with current LUB
    STR$COPY_DX,                                        ! Copy a string (LSET)
    STR$FREE_DX,                                        ! Free a string
    BAS$RSET,                                           ! Copy a string (RSET)
    BAS$$OPEN_ZERO : NOVALUE;                           ! Open channel 0

!+
! The following are the error codes used in this module.
!-

EXTERNAL LITERAL
    BAS$K_MAXMEMEXC : UNSIGNED (8),                     ! Maximum memory exceeded
    BAS$K_PROLOSSOR : UNSIGNED (8),                     ! Program lost, sorry
    BAS$K_IO_CHANOT : UNSIGNED (8),                     ! I/O channel not open
    BAS$K_ILLILLACC : UNSIGNED (8),                     ! Illegal or illogical access
    BAS$K_FIEOVEBUF : UNSIGNED (8),                     ! Field overflows buffer
    BAS$K_ILLFIEVAR : UNSIGNED (8),                     ! Illegal FIELD variable
    BAS$K_ILLIO_CHA : UNSIGNED (8);                     ! Illegal I/O channel

!<BLF/PAGE>
```

```
;    1 !+
;    1 ! The following field represents a symbol table entry.
;    1 !-
;    1
;    1 FIELD
;    1     BAS$FIELD_SYM =
;    1         SET
;    1         SYM$A_NEXT = [0, 0, %BPADDR, 0],           ! Next symbol table entry
;    1         SYM$A_PREV = [%UPVAL, 0, %BPADDR, 0],      ! Previous entry
;    1         SYM$L_CHAN = [%UPVAL*2, 0, %BPVAL, 0],     ! I/O channel
;    1         SYM$L_OFFSET = [%UPVAL*3, 0, %BPVAL, 0],           ! Offset into I/O buffer
;    1         SYM$L_LEN = [%UPVAL*4, 0, %BPVAL, 0],      ! Number of bytes referenced
;    1         SYM$L_DECL = [%UPVAL*5, 0, %BPVAL, 0],     ! Scope of the declaration
;    1         SYM$A_VAR = [%UPVAL*6, 0, %BPADDR, 0],     ! Address of descriptor
;    1         SYM$V_INVALID = [%UPVAL*7, 0, 1, 0]        ! "Invalid" bit
;    1         TES;
;    1
;    1 LITERAL
;    1     SYM$K_LENGTH = %UPVAL*8;                       ! Number of bytes to allocate
;    1
```

```
1 GLOBAL ROUTINE BAS$FIELD_SET                               ! Execute a FIELD statement
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Execute a FIELD statement.  The compiler pushes all of the
1 !       variables in the FIELD statement from right to left and then
1 !       calls this routine.  As a result, the formal parameters are
1 !       arranged rather strangely.  This routine goes through them
1 !       calling BAS$FIELD_VAR for each variable.
1 !
1 !       The FIELD statement is formatted as follows:
1 !
1 !       FIELD #chan, exp BY var, exp BY var, ...
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !   The formal parameters are rather strange, for the convenience of
1 !   the compiler.  Because the compiler likes to push parameters in the
1 !   order in which it encounters them, the pairs of optional parameters
1 !   are first, followed by the fixed parameters.  The list below is of
1 !   the parameters in reverse order.
1 !
1 !       CHAN.rl.v       An I/O channel.  Must be open.
1 !       DECL.rl.v       An indication of the scope of the declaration
1 !                       of the variable.  This is a pointer to the major
1 !                       frame (R11) if the variable is in the scope of
1 !                       the major procedure, or a pointer to the minor
1 !                       frame (R10) if the variable is in the scope of
1 !                       a DEF.
1 !
1 !   The following two parameters can be repeated as often as required.
1 !
1 !       LEN.rl.v        The number of bytes referenced by the variable
1 !       VAR.wt.d        The variable.  Since references to it ignore its
1 !                       previous (non-FIELD) contents, we free it here.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !       LUB$V_VA_USE    If this bit is set, the file has been used
1 !                       with a virtual array, and so cannot be used
1 !                       with the FIELD statement.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       SYM$Q_ROOT.mq
1 !       LUB$V_BLK_USE   This bit is set to prevent the file from
1 !                       being used with a virtual array.
1 !       LUB$V_FIELD_USE for this channel, set to 1
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
```

```
1  ! SIDE EFFECTS:
1  !
1  !       Adds symbols to the interpreter's symbol table, and/or modifies
1  !       symbols already there.
1  !
1  !--
1
2      BEGIN
2
2      GLOBAL REGISTER
2          CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2      BUILTIN
2          FP,
2          ACTUALCOUNT,
2          ACTUALPARAMETER;
2
2      LOCAL
2          FMP : REF BLOCK [, BYTE],
2          NUMARGS,
2          DECL,
2          CHAN,
2          LEN,
2          VAR,
2          OFFSET,
2          LUN_NO,
2          RBF,
2          RSZ;
2
2      FMP = .FP;
2  !+
2  ! Start at the beginning of the buffer.
2  !-
2      OFFSET = 0;
2  !+
2  ! Fetch, from the pecucliar argument list, the DECL and CHAN parameters.
2  !-
2      NUMARGS = ACTUALCOUNT ();
2      CHAN = ACTUALPARAMETER (.NUMARGS);
2      DECL = ACTUALPARAMETER (.NUMARGS - 1);
2  !+
2  ! Compute the logical unit number corresponding to th` channel, and
2  ! validate it.
2  !-
2
2      IF (.CHAN LSS 0) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
2
2      LUN_NO = (IF (.CHAN EQL 0) THEN LUB$K_LUN_INPU ELSE .CHAN);
2      BAS$$CB_PUSH (.LUN_NO, LUB$K_ILUN_MIN);
2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
2
3      IF ( NOT .CCB [LUB$V_OPENED])
2      THEN
3          IF  (.LUN_NO LSS 0)
2          THEN
3              BEGIN
3              BAS$$OPEN_ZERO (.FMP [SF$L_SAVE_FP] );
```

```
3              END
2          ELSE
3              BEGIN
3              BAS$$STOP_IO (BAS$K_IO_CHANOT);
2              END;
2
2      IF (.CCB [LUB$A_CLOSE] EQLA 0) THEN CCB [LUB$A_CLOSE] = BAS$$FIELD_KILL;
2
2      IF (.CCB [LUB$A_CLOSE] NEQA BAS$$FIELD_KILL) THEN BAS$$STOP_IO (BAS$K_ILLILLACC);
2
2      CCB [LUB$V_BLK_USE] = 1;
2
2      CCB [LUB$V_FIELD_USE] = 1;
2
2      IF (.CCB [LUB$V_VA_USE]) THEN BAS$$STOP_IO (BAS$K_ILLILLACC);
2
2  !+
2  ! Fetch the buffer address and length for later use
2  !-
2      RBF = .CCB [LUB$A_RBUF_ADR];
2      RSZ = .CCB [LUB$W_RBUF_SIZE];
2  !+
2  ! Go through the optional arguments, associating each variable with
2  ! its place in the I/O buffer.  We scan the variables from left to
2  ! right in the FIELD statement in case the same variable appears twice:
2  ! it should end up with its right-most association.
2  !-
2
2      DECR ARGNO FROM .NUMARGS - 2 TO 1 BY 2 DO
3          BEGIN
3
3          LOCAL
3              LEN,
3              VAR;
3
3          LEN = ACTUALPARAMETER (.ARGNO);
3          VAR = ACTUALPARAMETER (.ARGNO - 1);
3
3          IF (.OFFSET + .LEN GTR .RSZ) THEN BAS$$STOP_IO (BAS$K_FIEOVEBUF);
3
3          BAS$FIELD_VAR (.CHAN, .OFFSET, .LEN, .DECL, .VAR, .RBF);
3          OFFSET = .OFFSET + .LEN;
2          END;
2
2  !+
2  ! We are done with register CCB
2  !-
2      BAS$$CB_POP ();
2      RETURN;
1      END;                                            ! end of BAS$FIELD_SET

                                                         .TITLE  BAS$RSTS_FIELD
                                                         .IDENT  \1-023\

                                                         .PSECT  _BAS$DATA,NOEXE,  PIC,2
```

```
            00000000  00000000  00000 SYM$Q_ROOT:
                                              .LONG    0, 0                                          ;

                                              .EXTRN   BAS$$STOP, BAS$$STOP_IO
                                              .EXTRN   LIB$GET_VM, LIB$FREE_VM
                                              .EXTRN   BAS$$CB_PUSH, BAS$$CB_POP
                                              .EXTRN   BAS$$CB_GET, STR$COPY_DX
                                              .EXTRN   STR$FREE1_DX, BAS$RSET
                                              .EXTRN   BAS$$OPEN_ZERO, BAS$K_MAXMEMEXC
                                              .EXTRN   BAS$K_PROCOSSOR
                                              .EXTRN   BAS$K_IO_CHANOT
                                              .EXTRN   BAS$K_ILLILLACC
                                              .EXTRN   BAS$K_FIEOVEBUF
                                              .EXTRN   BAS$K_ILLFIEVAR
                                              .EXTRN   BAS$K_ILLIO_CHA

                                              .PSECT   _BAS$CODE,NOWRT,  SHR,  PIC,2

                           0FFC 00000         .ENTRY   BAS$FIELD_SET, Save R2,R3,R4,R5,R6,R7,R8,-   ; 1234
                                                       R9,R10,R1T                                    ;
           5A 00000000G   00 9E 00002         MOVAB    BAS$$STOP_IO, R10                             ;
           55               5D D0 00009         MOVL     FP, FMP                                       ; 1320
                            54 D4 0000C         CLRL     OFFSET                                        ; 1324
           53               6C 9A 0000E         MOVZBL   (AP), NUMARGS                                 ; 1328
           56             6C43 D0 00011         MOVL     (AP)[NUMARGS], CHAN                           ; 1329
           59        FC AC43 D0 00015         MOVL     -4(AP)[NUMARGS], DECL                         ; 1330
                            56 D5 0001A         TSTL     CHAN                                          ; 1336
                            0B 18 0001C         BGEQ     1$                                            ;
           7E       00G   8F 9A 0001E         MOVZBL   #BAS$K_ILLIO_CHA, -(SP)                       ;
00000000G  00             01 FB 00022         CALLS    #1, BAS$$STOP                                 ;
                            56 D5 00029 1$:     TSTL     CHAN                                          ; 1338
                            05 12 0002B         BNEQ     2$                                            ;
           52               07 CE 0002D         MNEGL    #7, LUN_NO                                    ;
                            03 11 00030         BRB      3$                                            ;
           52               56 D0 00032 2$:     MOVL     CHAN, LUN_NO                                  ;
           50               08 CE 00035 3$:     MNEGL    #8, R0                                        ; 1339
              00000000G   00 16 00038         JSB      BAS$$CB_PUSH                                  ;
     FF4C  CB        0C   A5 D0 0003E         MOVL     12(FMP), -180(CCB)                            ; 1340
           17        FC   AB E8 00044         BLBS     -4(CCB), 5$                                   ; 1342
                            52 D5 00048         TSTL     LUN_NO                                        ; 1344
                            0C 18 0004A         BGEQ     4$                                            ;
                     0C   A5 DD 0004C         PUSHL    12(FMP)                                       ; 1347
00000000G  00             01 FB 0004F         CALLS    #1, BAS$$OPEN_ZERO                            ;
                            07 11 00056         BRB      5$                                            ; 1344
           7E       00G   8F 9A 00058 4$:     MOVZBL   #BAS$K_IO_CHANOT, -(SP)                       ; 1351
           6A             01 FB 0005C         CALLS    #1, BAS$$STOP_IO                              ;
                     A4   AB D5 0005F 5$:     TSTL     -92(CCB)                                      ; 1354
                            06 12 00062         BNEQ     6$                                            ;
       A4  AB     0000V   CF 9E 00064         MOVAB    BAS$$FIELD_KILL, -92(CCB)                     ;
           50     0000V   CF 9E 0006A 6$:     MOVAB    BAS$$FIELD_KILL, R0                           ; 1356
           50        A4   AB D1 0006F         CMPL     -92(CCB), R0                                  ;
                            07 13 00073         BEQL     7$                                            ;
           7E       00G   8F 9A 00075         MOVZBL   #BAS$K_ILLILLACC, -(SP)                       ;
           6A             01 FB 00079         CALLS    #1, BAS$$STOP_IO                              ;
       FF  AB             02 88 0007C 7$:     BISB2    #2, -1(CCB)                                   ; 1358
       A1  AB        40   8F 88 00080         BISB2    #64, -95(CCB)                                 ; 1360
           07        FF   AB E9 00085         BLBC     -1(CCB), 8$                                   ; 1362
```

```
                    7E        00G  8F   9A 00089           MOVZBL  #BAS$K_ILLILLACC, -(SP)                     ;
                    6A             01   FB 0008D           CALLS   #1, BAS$$STOP_IO                            ;
                    58        EC   AB   D0 00090  8$:      MOVL    -20(CCB), RBF                               ; 1367
                    57        D2   AB   3C 00094           MOVZWL  -46(CCB), RSZ                               ; 1368
                    52             53   D0 00098           MOVL    NUMARGS, ARGNO                              ; 1388
                                   2D   11 0009B           BRB     11$                                         ;
                    53           6C42   D0 0009D  9$:      MOVL    (AP)[ARGNO], LEN                            ; 1383
                    55        FC AC42   D0 000A1           MOVL    -4(AP)[ARGNO], VAR                          ; 1384
          50        54             53   C1 000A6           ADDL3   LEN, OFFSET, R0                             ; 1386
                    57             50   D1 000AA           CMPL    R0, RSZ                                     ;
                                   07   15 000AD           BLEQ    10$                                         ;
                    7E        00G  8F   9A 000AF           MOVZBL  #BAS$K_FIEOVEBUF, -(SP)                     ;
                    6A             01   FB 000B3           CALLS   #1, BAS$$STOP_IO                            ;
                            0120   8F   BB 000B6  10$:     PUSHR   #^M<R5,R8>                                  ; 1388
                            0208   8F   BB 000BA           PUSHR   #^M<R3,R9>                                  ;
                                   54   DD 000BE           PUSHL   OFFSET                                      ;
                                   56   DD 000C0           PUSHL   CHAN                                        ;
               0000V  CF           06   FB 000C2           CALLS   #6, BAS$FIELD_VAR                           ;
                      54           53   C0 000C7           ADDL2   LEN, OFFSET                                 ; 1389
                      52           02   C2 000CA  11$:     SUBL2   #2, ARGNO                                   ; 1376
                                   CE   14 000CD           BGTR    9$                                          ;
                        00000000G  00   16 000CF           JSB     BAS$$CB_POP                                 ; 1395
                                        04 000D5           RET                                                 ; 1397

; Routine Size:  214 bytes,    Routine Base:  _BAS$CODE + 0000


;  371          1398  1
```

```
1 ROUTINE BAS$FIELD_VAR (                                  ! "Declare" a field variable
1         CHAN,                                            ! The I/O channel whose buffer this variable references
1         OFFSET,                                          ! The offset in the buffer
1         LEN,                                             ! The number of bytes referenced
1         DECL,                                            ! Scope of the declaration
1         VAR,                                             ! Address of this variable's descriptor
1         RBF                                              ! Start of the record buffer
1     ) : CALL_CCB NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       "Declares" a field variable.  Such a variable refers to the
1 !       buffer of an I/O channel.  To avoid leaving obsolete addresses
1 !       in a user's program each reference to a FIELD variable is
1 !       interpreted.  This routine puts the variable in the interpreter
1 !       symbol table so it can be found by BAS$FIELD_COPY.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       CHAN.rl.v       An I/O channel.  Need not be open yet.
1 !       OFFSET.rl.v     The offset into that channel's buffer of the
1 !                       start of the area referenced by the variable
1 !       LEN.rl.v        The number of bytes referenced by the variable
1 !       DECL.rl.v       An indication of the scope of the declaration
1 !                       of the variable.  This is a pointer to the major
1 !                       frame (R11) if the variable is in the scope of
1 !                       the major procedure, or a pointer to the minor
1 !                       frame (R10) if the variable is in the scope of
1 !                       a DEF.
1 !       VAR.wt.d        The variable.  Its storage is freed and it is
1 !                       made to point to the buffer, so the compiled
1 !                       code can do read accesses through it.
1 !       RBF.ra.v        Address of the file's record buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       SYM$Q_ROOT.mq
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Adds a symbol to the interpreter's symbol table, or modifies one
1 !       already there.
1 !
1 !--
1
2     BEGIN
2
```

```
2       EXTERNAL REGISTER
2           CCB : REF BLOCK [, BYTE];
2
2       MAP
2           VAR : REF BLOCK [8, BYTE];
2
2       LOCAL
2           SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2           SEARCH_DONE;
2
2   !+
2   ! If the symbol table root has not yet been initialized, initialize it.
2   !-
2
3       IF (.SYM$Q_ROOT [0] EQL 0)
2       THEN
3           BEGIN
3
3           LOCAL
3               AST_STATUS;
3
3           AST_STATUS = $SETAST (ENBFLG = 0);
3
4           IF (.SYM$Q_ROOT [0] EQL 0)
3           THEN
4               BEGIN
4               SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
3               END;
3
3           IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
3
2           END;
2
2   !+
2   ! Search the queue to see if this variable is already on it.
2   !-
2       SYM = .SYM$Q_ROOT [0];
2       SEARCH_DONE = 0;
2
2       DO
3           BEGIN
3
4           IF (.SYM EQLA SYM$Q_ROOT)
3           THEN
3               SEARCH_DONE = 1
3           ELSE
3
4               IF (.SYM [SYM$A_VAR] EQLA .VAR)
3               THEN
4                   BEGIN
4
4                   IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP_IO (BAS$K_ILLFIEVAR);
4
4                   SEARCH_DONE = 3;
3                   END;
3
3           IF ( NOT .SEARCH_DONE) THEN SYM = .SYM [SYM$A_NEXT];
```

```
3
3           END
2       UNTIL (.SEARCH_DONE);
2
3       IF (.SEARCH_DONE EQL 1)
2       THEN
3           BEGIN
3   !+
3   ! We must create a symbol table entry.
3   !-
3
3           BUILTIN
3               INSQUE;
3
3           LOCAL
3               GET_VM_STATUS,
3               INSQUE_ADDR;
3
3           GET_VM_STATUS = LIB$GET_VM (%REF (SYM$K_LENGTH), SYM);
3
3           IF ( NOT .GET_VM_STATUS) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
3
3           INSQUE_ADDR = SYM$Q_ROOT [1];
3           INSQUE (.SYM, ..INSQUE_ADDR);              ! Tail of queue
3   !+
3   ! Make sure the string is empty.
3   !-
3           STR$FREE1_DX (.VAR);
2           END;
2
2   !+
2   ! Fill in the symbol table entry
2   !-
2       SYM [SYM$L_CHAN] = .CHAN;
2       SYM [SYM$L_OFFSET] = .OFFSET;
2       SYM [SYM$L_LEN] = .LEN;
2       SYM [SYM$L_DECL] = .DECL;
2       SYM [SYM$A_VAR] = .VAR;
2       SYM [SYM$V_INVALID] = 0;
2       VAR [DSC$W_LENGTH] = MAX (0, .LEN);
2       VAR [DSC$B_CLASS] = DSC$K_CLASS_S;
2       VAR [DSC$A_POINTER] = .RBF + .OFFSET;
2       RETURN;
1       END;                                           ! end of BAS$FIELD_VAR


                                            .EXTRN  SYS$SETAST

                       007C 00000 BAS$FIELD_VAR:
                                            .WORD   Save R2,R3,R4,R5,R6                    ; 1399
   56 00000000G   00   9E 00002             MOVAB   BAS$$STOP_IO, R6                       ;
   55 00000000G   00   9E 00009             MOVAB   SYS$SETAST, R5                         ;
   54 00000000'   EF   9E 00010             MOVAB   SYM$Q_ROOT, R4                         ;
   5E             08   C2 00017             SUBL2   #8, SP                                 ;
                  64   D5 0001A             TSTL    SYM$Q_ROOT                             ; 1470
                  1D   12 0001C             BNEQ    2$                                     ;
```

```
                                 7E D4 0001E       CLRL     -(SP)                             ; 1477
                    65           01 FB 00020       CALLS    #1, SYS$SETAST                    ;
                                 64 D5 00023       TSTL     SYM$Q_ROOT                        ; 1479
                                 0A 12 00025       BNEQ     1$                                ;
                    51           64 9E 00027       MOVAB    SYM$Q_ROOT, R1                    ; 1482
                04  A4           51 D0 0002A       MOVL     R1, SYM$Q_ROOT+4                  ;
                    64           51 D0 0002E       MOVL     R1, SYM$Q_ROOT                    ;
                    09           50 D1 00031 1$:   CMPL     AST_STATUS, #9                    ; 1485
                                 05 12 00034       BNEQ     2$                                ;
                                 01 DD 00036       PUSHL    #1                                ;
                    65           01 FB 00038       CALLS    #1, SYS$SETAST                    ;
                04  AE           64 D0 0003B 2$:   MOVL     SYM$Q_ROOT, SYM                   ; 1492
                                 53 D4 0003F       CLRL     SEARCH_DONE                       ; 1493
                    52      04   AE D0 00041 3$:   MOVL     SYM, R2                           ; 1498
                    50           64 9E 00045       MOVAB    SYM$Q_ROOT, R0                    ;
                    50           52 D1 00048       CMPL     R2, R0                            ;
                                 05 12 0004B       BNEQ     4$                                ;
                    53           01 D0 0004D       MOVL     #1, SEARCH_DONE                   ; 1500
                                 15 11 00050       BRB      6$                                ;
                14  AC      18   A2 D1 00052 4$:   CMPL     24(R2), VAR                       ; 1503
                                 0E 12 00057       BNEQ     6$                                ;
                    07      1C   A2 E9 00059       BLBC     28(R2), 5$                        ; 1507
                    7E     00G   8F 9A 0005D       MOVZBL   #BAS$K_ILLFIEVAR, -(SP)           ;
                    66           01 FB 00061       CALLS    #1, BAS$$STOP_IO                  ;
                    53           03 D0 00064 5$:   MOVL     #3, SEARCH_DONE                   ; 1509
                    07           53 E8 00067 6$:   BLBS     SEARCH_DONE, 7$                   ; 1512
                04  AE           62 D0 0006A       MOVL     (R2), SYM                         ;
                    D0           53 E9 0006E       BLBC     SEARCH_DONE, 3$                   ; 1515
                    01           53 D1 00071 7$:   CMPL     SEARCH_DONE, #1                   ; 1517
                                 2E 12 00074       BNEQ     9$                                ;
                            04   AE 9F 00076       PUSHAB   SYM                               ; 1531
                04  AE           20 D0 00079       MOVL     #32, 4(SP)                        ;
                            04   AE 9F 0007D       PUSHAB   4(SP)                             ;
         00000000G  00           02 FB 00080       CALLS    #2, LIB$GET_VM                    ;
                    07           50 E8 00087       BLBS     GET_VM_STATUS, 8$                 ; 1533
                    7E     00G   8F 9A 0008A       MOVZBL   #BAS$K_MAXMEMEXC, -(SP)           ;
                    66           01 FB 0008E       CALLS    #1, BAS$$STOP_IO                  ;
                    50      04   A4 9E 00091 8$:   MOVAB    SYM$Q_ROOT+4, INSQUE_ADDR         ; 1535
                0C  B0      04   BE 0E 00095       INSQUE   @SYM, @0(INSQUE_ADDR)             ; 1536
                            14   AC DD 0009A       PUSHL    VAR                               ; 1540
         00000000G  00           01 FB 0009D       CALLS    #1, STR$FREE1_DX                  ;
                    50      04   AE D0 000A4 9$:   MOVL     SYM, R0                           ; 1546
                08  A0      04   AC 7D 000A8       MOVQ     CHAN, 8(R0)                       ;
                10  A0      0C   AC 7D 000AD       MOVQ     LEN, 16(R0)                       ; 1548
                    51      14   AC D0 000B2       MOVL     VAR, R1                           ; 1550
                18  A0           51 D0 000B6       MOVL     R1, 24(R0)                        ;
                1C  A0           01 8A 000BA       BICB2    #1, 28(R0)                        ; 1551
                    50      0C   AC D0 000BE       MOVL     LEN, R0                           ; 1552
                                 02 18 000C2       BGEQ     10$                               ;
                                 50 D4 000C4       CLRL     R0                                ;
                    61           50 B0 000C6 10$:  MOVW     R0, (R1)                          ;
                03  A1           01 90 000C9       MOVB     #1, 3(R1)                         ; 1553
04    A1        18  AC      08   AC C1 000CD       ADDL3    OFFSET, RBF, 4(R1)                ; 1554
                                    04 000D4       RET                                        ; 1556

; Routine Size:  213 bytes,    Routine Base:  _BAS$CODE + 00D6
;  531          1557  1
```

```
1 GLOBAL ROUTINE BAS$FIELD_CLEAR (                       ! Undeclare a field variable
1           VAR                                         ! The variable being cleared
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Undeclare a possible FIELD variable.  This routine is called
1 !         prior to any BASIC statement that causes a field variable to
1 !         lose its FIELD attribute, if that variable has a FIELD
1 !         statement associated with it anywhere in the program.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         VAR.at.d          The variable.  Only the address of its
1 !                           descriptor is used, to scan the symbol table.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         SYM$Q_ROOT.mq     The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         SYM$Q_ROOT.mq
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         May remove a symbol from the symbol table.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         VAR : REF BLOCK [8, BYTE];
2
2     LOCAL
2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2         SEARCH_DONE;
2
2 !+
2 ! If the symbol table root has not yet been initialized, initialize it.
2 !-
2
3     IF (.SYM$Q_ROOT [0] EQL 0)
2     THEN
3         BEGIN
3
3         LOCAL
3             AST_STATUS;
3
3         AST_STATUS = $SETAST (ENBFLG = 0);
```

```
3
4           IF (.SYM$Q_ROOT [0] EQL 0)
3           THEN
4               BEGIN
4               SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
3               END;
3
3           IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
3
2           END;
2
2   !+
2   ! Search the symbol table, removing this variable if it is present.
2   !-
2       SYM = .SYM$Q_ROOT [0];
2       SEARCH_DONE = 0;
2
2       DO
3           BEGIN
3
4           IF (.SYM EQLA SYM$Q_ROOT)
3           THEN
3               SEARCH_DONE = 1
3           ELSE
3
4               IF (.SYM [SYM$A_VAR] EQL .VAR)
3               THEN
4                   BEGIN
4   !+
4   ! We must delete this symbol from the symbol table.
4   !-
4
4                   BUILTIN
4                       REMQUE;
4
4                   LOCAL
4                       FREE_VM_STATUS,
4                       TEMP;
4
4                   IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
4
4                   REMQUE (.SYM, TEMP);
4                   VAR [DSC$W_LENGTH] = 0;
4                   VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
4                   VAR [DSC$A_POINTER] = 0;
4                   FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
4
4                   IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
4
4                   SEARCH_DONE = 1
4                   END
3               ELSE
3                   SYM = .SYM [SYM$A_NEXT];
3
3           END
2       UNTIL (.SEARCH_DONE);
2
```

```
;  647          1672  1     END;                                    ! end of BAS$FIELD_CLEAR



                                  007C 00000          .ENTRY   BAS$FIELD_CLEAR, Save R2,R3,R4,R5,R6          ; 1558
                   56 00000000G  00  9E 00002          MOVAB    BAS$$STOP, R6                                  ;
                   55 00000000G  00  9E 00009          MOVAB    SYS$SETAST, R5                                 ;
                   54 00000000'  EF  9E 00010          MOVAB    SYM$Q_ROOT, R4                                 ;
                   5E            08  C2 00017          SUBL2    #8, SP                                         ;
                                 64  D5 0001A          TSTL     SYM$Q_ROOT                                     ; 1607
                                 1D  12 0001C          BNEQ     2$                                             ;
                                 7E  D4 0001E          CLRL     -(SP)                                          ; 1614
                   65            01  FB 00020          CALLS    #1, SYS$SETAST                                 ;
                                 64  D5 00023          TSTL     SYM$Q_ROOT                                     ; 1616
                                 0A  12 00025          BNEQ     1$                                             ;
                   51            64  9E 00027          MOVAB    SYM$Q_ROOT, R1                                 ; 1619
              04   A4            51  D0 0002A          MOVL     R1, SYM$Q_ROOT+4                               ;
                   64            51  D0 0002E          MOVL     R1, SYM$Q_ROOT                                 ;
                   09            50  D1 00031 1$:      CMPL     AST_STATUS, #9                                 ; 1622
                                 05  12 00034          BNEQ     2$                                             ;
                                 01  DD 00036          PUSHL    #1                                             ;
                   65            01  FB 00038          CALLS    #1, SYS$SETAST                                 ;
                   52            64  D0 0003B 2$:      MOVL     SYM$Q_ROOT, SYM                                ; 1629
                                 53  D4 0003E          CLRL     SEARCH_DONE                                    ; 1630
                   50            64  9E 00040 3$:      MOVAB    SYM$Q_ROOT, R0                                 ; 1635
                   50            52  D1 00043          CMPL     SYM, R0                                        ;
                                 3E  13 00046          BEQL     5$                                             ;
              04   AC       18   A2  D1 00048          CMPL     24(SYM), VAR                                   ; 1640
                                 3C  12 0004D          BNEQ     6$                                             ;
                   07       1C   A2  E9 0004F          BLBC     28(SYM), 4$                                    ; 1654
                   7E      00G   8F  9A 00053          MOVZBL   #BAS$K_ILLFIEVAR, -(SP)                        ;
                   66            01  FB 00057          CALLS    #1, BAS$$STOP                                  ;
              04   AE            62  0F 0005A 4$:      REMQUE   (SYM), TEMP                                    ; 1656
                   50       04   AC  D0 0005E          MOVL     VAR, R0                                        ; 1657
                                 60  B4 00062          CLRW     (R0)                                           ;
              03   A0            02  90 00064          MOVB     #2, 3(R0)                                      ; 1658
                            04   A0  D4 00068          CLRL     4(R0)                                          ; 1659
                            04   AE  9F 0006B          PUSHAB   TEMP                                           ; 1660
              04   AE            20  D0 0006E          MOVL     #32, 4(SP)                                     ;
                            04   AE  9F 00072          PUSHAB   4(SP)                                          ;
       00000000G   00            02  FB 00075          CALLS    #2, LIB$FREE_VM                                ;
                   07            50  E8 0007C          BLBS     FREE_VM_STATUS, 5$                             ; 1662
                   7E      00G   8F  9A 0007F          MOVZBL   #BAS$K_PROLOSSOR, -(SP)                        ;
                   66            01  FB 00083          CALLS    #1, BAS$$STOP                                  ;
                   53            01  D0 00086 5$:      MOVL     #1, SEARCH_DONE                                ; 1664
                                 03  11 00089          BRB      7$                                             ;
                   52            62  D0 0008B 6$:      MOVL     (SYM), SYM                                     ; 1667
                   AF            53  E9 0008E 7$:      BLBC     SEARCH_DONE, 3$                                ; 1670
                                     04 00091          RET                                                     ; 1672

; Routine Size:  146 bytes,    Routine Base:  _BAS$CODE + 01AB


;  648          1673  1
```

```
1 GLOBAL ROUTINE BAS$FIELD_COPY (                          ! Copy to or from a FIELD variable
1           STMT_TYPE,                                     ! Either LSET or RSET
1           VAR2,                                          ! The destination variable
1           VAR1                                           ! The source variable
1       ) : NOVALUE =
1
1   !++
1   ! FUNCTIONAL DESCRIPTION:
1   !
1   !       Copies between two string variables.  One or the other may
1   !       by FIELD variables, but not both.  Because the compiler cannot
1   !       be sure if a FIELD statement has been issued to a variable
1   !       (since it cannot trace program flow) it is possible that
1   !       neither variable is FIELD.
1   !
1   ! FORMAL PARAMETERS:
1   !
1   !       STMT_TYPE.rl.v  0 = this is an LSET statement, 1 = RSET
1   !       VAR2.wt.dx      The destination of the copy.  This may be a
1   !                       FIELD variable.
1   !       VAR1.rt.dx      The source for the copy.  This may be a FIELD
1   !                       variable.
1   ! IMPLICIT INPUTS:
1   !
1   !       SYM$Q_ROOT.rq   The queue of FIELD variables : the symbol table.
1   !
1   ! IMPLICIT OUTPUTS:
1   !
1   !       NONE
1   !
1   ! ROUTINE VALUE:
1   ! COMPLETION CODES:
1   !
1   !       NONE
1   !
1   ! SIDE EFFECTS:
1   !
1   !       May write into or read from an I/O buffer.
1   !
1   !--
1
2       BEGIN
2
2       BUILTIN
2           FP;
2
2       GLOBAL REGISTER
2           CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2       MAP
2           VAR1 : REF BLOCK [8, BYTE],
2           VAR2 : REF BLOCK [8, BYTE];
2
2       LOCAL
2           FMP : REF BLOCK [, BYTE],
2           SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2           SEARCH_DONE,
;  2          VAR1_DESC : BLOCK [8, BYTE],
;  2          VAR2_DESC : BLOCK [8, BYTE],
;  2          VAR1_DESC_ADR : REF BLOCK [8, BYTE],
;  2          VAR2_DESC_ADR : REF BLOCK [8, BYTE],
;  2          VAR1_CHAN,
;  2          VAR2_CHAN;
;  2
;  2      FMP = .FP;
;  2  !+
;  2  ! If the symbol table root has not yet been initialized, initialize it.
;  2  !-
;  2
;  3      IF (.SYM$Q_ROOT [0] EQL 0)
;  2      THEN
;  3          BEGIN
;  3
;  3          LOCAL
;  3              AST_STATUS;
;  3
;  3          AST_STATUS = $SETAST (ENBFLG = 0);
;  3
;  4          IF (.SYM$Q_ROOT [0] EQL 0)
;  3          THEN
;  4              BEGIN
;  4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  3              END;
;  3
;  3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  3
;  2          END;
;  2
;  2  !+
;  2  ! Search the queue to see if the input variable is on it.
;  2  !-
;  2      SYM = .SYM$Q_ROOT [0];
;  2      SEARCH_DONE = 0;
;  2
;  2      DO
;  3          BEGIN
;  3
;  4          IF (.SYM EQLA SYM$Q_ROOT)
;  3          THEN
;  3              SEARCH_DONE = 1
;  3          ELSE
;  3
;  3              IF (.SYM [SYM$A_VAR] EQLA .VAR1) THEN SEARCH_DONE = 3;
;  3
;  3          IF ( NOT .SEARCH_DONE) THEN SYM = .SYM [SYM$A_NEXT];
;  3
;  3          END
;  2      UNTIL (.SEARCH_DONE);
;  2
;  3      IF (.SEARCH_DONE EQL 1)
;  2      THEN
;  3          BEGIN
;  3  !+
;  3  ! The variable is not in the symbol table.  That must mean that it
```

```
3 ! is not a FIELD variable.
3 !-
3         VAR1_DESC_ADR = .VAR1;
3         VAR1_CHAN = 0;
3         END
2     ELSE
3         BEGIN
3 !+
3 ! Don't touch a variable marked invalid.
3 !-
3
3         IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
3
3 !+
3 ! The variable is in the symbol table.  Construct a descriptor for it.
3 !-
3         VAR1_DESC_ADR = VAR1_DESC;
3         VAR1_DESC [DSC$W_LENGTH] = MAX (0, .SYM [SYM$L_LEN]);
3         VAR1_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
3         VAR1_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
3         VAR1_CHAN = .SYM [SYM$L_CHAN];
3
3         IF (.VAR1_CHAN EQL 0) THEN VAR1_CHAN = LUB$K_LUN_INPU;
3
3         BAS$$CB_PUSH (.VAR1_CHAN, LUB$K_LUN_INPU);
3         CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
3         VAR1_DESC [DSC$A_POINTER] = .CCB [LUB$A_RBUF_ADR] + .SYM [SYM$L_OFFSET];
3
3         IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP (BAS$K_IO_CHANOT);
3
4         IF (.CCB [LUB$W_RBUF_SIZE] LSSU .SYM [SYM$L_OFFSET] + .SYM [SYM$L_LEN])
3         THEN
3             BAS$$STOP_IO (BAS$K_FIEOVEBUF);
3
2         END;
2
2 !+
2 ! Search the queue to see if the output variable is on it.
2 !-
2     SYM = .SYM$Q_ROOT [0];
2     SEARCH_DONE = 0;
2
2     DO
3         BEGIN
3
4         IF (.SYM EQLA SYM$Q_ROOT)
3         THEN
3             SEARCH_DONE = 1
3         ELSE
3
3             IF (.SYM [SYM$A_VAR] EQLA .VAR2) THEN SEARCH_DONE = 3;
3
3         IF ( NOT .SEARCH_DONE) THEN SYM = .SYM [SYM$A_NEXT];
3
3         END
2     UNTIL (.SEARCH_DONE);
2
```

```
3        IF (.SEARCH_DONE EQL 1)
2        THEN
3            BEGIN
3    !+
3    ! The variable is not in the symbol table.  That must mean that it
3    ! is not a FIELD variable.
3    !-
3            VAR2_DESC_ADR = .VAR2;
3            VAR2_CHAN = 0;
3            END
2        ELSE
3            BEGIN
3
3            IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
3
3    !+
3    ! The variable is in the symbol table.  Construct a descriptor for it.
3    !-
3            VAR2_DESC_ADR = VAR2_DESC;
3            VAR2_DESC [DSC$W_LENGTH] = MAX (0, .SYM [SYM$L_LEN]);
3            VAR2_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
3            VAR2_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
3            VAR2_CHAN = .SYM [SYM$L_CHAN];
3
3            IF (.VAR2_CHAN EQL 0) THEN VAR2_CHAN = LUB$K_LUN_INPU;
3
3            BAS$$CB_PUSH (.VAR2_CHAN, LUB$K_LUN_INPU);
3            CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
3            VAR2_DESC [DSC$A_POINTER] = .CCB [LUB$A_RBUF_ADR] + .SYM [SYM$L_OFFSET];
3
3            IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP (BAS$K_IO_CHANOT);
3
4            IF (.CCB [LUB$W_RBUF_SIZE] LSSU .SYM [SYM$L_OFFSET] + .SYM [SYM$L_LEN])
3            THEN
3                BAS$$STOP_IO (BAS$K_FIEOVEBUF);
3
2            END;
2
2    !+
2    ! Copy from the input variable to the output variable.
2    ! We must observe the semantics of the statement type.
2    !-
2
2        CASE .STMT_TYPE FROM STMT_TYPE_LSET TO STMT_TYPE_RSET OF
2            SET
2
2            [STMT_TYPE_LSET] :
2                STR$COPY_DX (.VAR2_DESC_ADR, .VAR1_DESC_ADR);
2
2            [STMT_TYPE_RSET] :
2                BAS$RSET (.VAR2_DESC_ADR, .VAR1_DESC_ADR);
2            TES;
2
2    !+
2    ! Release register CCB
2    !-
2
```

```
;  878       1902 3     IF (.VAR2_CHAN NEQ 0)
;  879       1903 2     THEN
;  880       1904 3         BEGIN
;  881       1905 3         BAS$$CB_GET ();
;  882       1906 3         BAS$$CB_POP ();
;  883       1907 2         END;
;  884       1908 2
;  885       1909 3     IF (.VAR1_CHAN NEQ 0)
;  886       1910 2     THEN
;  887       1911 3         BEGIN
;  888       1912 3         BAS$$CB_GET ();
;  889       1913 3         BAS$$CB_POP ();
;  890       1914 2         END;
;  891       1915 2
;  892       1916 1     END;                                    ! end of BAS$FIELD_COPY



                           0FFC 00000          .ENTRY  BAS$FIELD_COPY, Save R2,R3,R4,R5,R6,R7,R8,- ; 1674
                                                       R9,R10,R11                                  ;
         5A 00000000G  00   9E 00002          MOVAB   SYS$SETAST, R10                             ;
         59 00000000G  00   9E 00009          MOVAB   BAS$$STOP, R9                               ;
         58 00000000'  EF   9E 00010          MOVAB   SYM$Q_ROOT, R8                              ;
         5E            10   C2 00017          SUBL2   #16, SP                                     ;
         55            5D   D0 0001A          MOVL    FP, FMP                                     ; 1738
                       68   D5 0001D          TSTL    SYM$Q_ROOT                                  ; 1743
                       1D   12 0001F          BNEQ    2$                                          ;
                       7E   D4 00021          CLRL    -(SP)                                       ; 1750
         6A            01   FB 00023          CALLS   #1, SYS$SETAST                              ;
                       68   D5 00026          TSTL    SYM$Q_ROOT                                  ; 1752
                       0A   12 00028          BNEQ    1$                                          ;
         51            68   9E 0002A          MOVAB   SYM$Q_ROOT, R1                              ; 1755
   04    A8            51   D0 0002D          MOVL    R1, SYM$Q_ROOT+4                            ;
         68            51   D0 00031          MOVL    R1, SYM$Q_ROOT                              ;
         09            50   D1 00034 1$:      CMPL    AST_STATUS, #9                              ; 1758
                       05   12 00037          BNEQ    2$                                          ;
                       01   DD 00039          PUSHL   #1                                          ;
         6A            01   FB 0003B          CALLS   #1, SYS$SETAST                              ;
         53            68   D0 0003E 2$:      MOVL    SYM$Q_ROOT, SYM                             ; 1765
                       54   D4 00041          CLRL    SEARCH_DONE                                 ; 1766
         50            68   9E 00043 3$:      MOVAB   SYM$Q_ROOT, R0                              ; 1771
         50            53   D1 00046          CMPL    SYM, R0                                     ;
                       05   12 00049          BNEQ    4$                                          ;
         54            01   D0 0004B          MOVL    #1, SEARCH_DONE                             ; 1773
                       0A   11 0004E          BRB     5$                                          ;
   0C    AC       18   A3   D1 00050 4$:      CMPL    24(SYM), VAR1                               ; 1776
                       03   12 00055          BNEQ    5$                                          ;
         54            03   D0 00057          MOVL    #3, SEARCH_DONE                             ;
         06            54   E8 0005A 5$:      BLBS    SEARCH_DONE, 6$                             ; 1778
         53            63   D0 0005D          MOVL    (SYM), SYM                                  ;
         E0            54   E9 00060          BLBC    SEARCH_DONE, 3$                             ; 1781
         01            54   D1 00063 6$:      CMPL    SEARCH_DONE, #1                             ; 1783
                       08   12 00066          BNEQ    7$                                          ;
         57       0C   AC   D0 00068          MOVL    VAR1, VAR1_DESC_ADR                         ; 1790
                       56   D4 0006C          CLRL    VAR1_CHAN                                   ; 1791
```

```
                                              67 11 0006E       BRB     12$                                 ; 1783
                                  07          1C  A3 E9 00070 7$:   BLBC    28(SYM), 8$                         ; 1799
                                  7E         00G  8F 9A 00074       MOVZBL  #BAS$K_ILLFIEVAR, -(SP)             ;
                                  69              01 FB 00078       CALLS   #1, BAS$$STOP                       ;
                                  57          08  AE 9E 0007B 8$:   MOVAB   VAR1_DESC, VAR1_DESC_ADR            ; 1804
                                  50          10  A3 D0 0007F       MOVL    16(SYM), R0                         ; 1805
                                                  02 18 00083       BGEQ    9$                                  ;
                                                  50 D4 00085       CLRL    R0                                  ;
                          08  AE                  50 B0 00087 9$:   MOVW    R0, VAR1_DESC                       ;
                          0A  AE        010E  8F B0 0008B       MOVW    #270, VAR1_DESC+2                   ; 1806
                                  56          08  A3 D0 00091       MOVL    8(SYM), VAR1_CHAN                   ; 1808
                                                  03 12 00095       BNEQ    10$                                 ; 1810
                                  56              07 CE 00097       MNEGL   #7, VAR1_CHAN                       ;
                                  50              07 CE 0009A 10$:  MNEGL   #7, R0                              ; 1812
                                  52              56 D0 0009D       MOVL    VAR1_CHAN, R2                       ;
                                       00000000G  00 16 000A0       JSB     BAS$$CB_PUSH                        ;
                        FF4C  CB          0C  A5 D0 000A6       MOVL    12(FMP), -180(CCB)                  ; 1813
        0C    AE          EC  AB          0C  A3 C1 000AC       ADDL3   12(SYM), -20(CCB), VAR1_DESC+4      ; 1814
                                  07          FC  AB E8 000B3       BLBS    -4(CCB), 11$                        ; 1816
                                  7E         00G  8F 9A 000B7       MOVZBL  #BAS$K_IO_CHANOT, -(SP)             ;
                                  69              01 FB 000BB       CALLS   #1, BAS$$STOP                       ;
              50          0C  A3          10  A3 C1 000BE 11$:  ADDL3   16(SYM), 12(SYM), R0                ; 1818
  50    D2    AB              10              00 ED 000C4       CMPZV   #0, #16, -46(CCB), R0               ;
                                                  0B 1E 000CA       BGEQU   12$                                 ;
                                  7E         00G  8F 9A 000CC       MOVZBL  #BAS$K_FIEOVEBUF, -(SP)             ; 1820
                   00000000G  00              01 FB 000D0       CALLS   #1, BAS$$STOP_IO                    ;
                                  53              68 D0 000D7 12$:  MOVL    SYM$Q_ROOT, SYM                     ; 1827
                                                  54 D4 000DA       CLRL    SEARCH_DONE                         ; 1828
                                  50              68 9E 000DC 13$:  MOVAB   SYM$Q_ROOT, R0                      ; 1833
                                  50              53 D1 000DF       CMPL    SYM, R0                             ;
                                                  05 12 000E2       BNEQ    14$                                 ;
                                  54              01 D0 000E4       MOVL    #1, SEARCH_DONE                     ; 1835
                                                  0A 11 000E7       BRB     15$                                 ;
                          08  AC          18  A3 D1 000E9 14$:  CMPL    24(SYM), VAR2                       ; 1838
                                                  03 12 000EE       BNEQ    15$                                 ;
                                  54              03 D0 000F0       MOVL    #3, SEARCH_DONE                     ;
                                  06              54 E8 000F3 15$:  BLBS    SEARCH_DONE, 16$                    ; 1840
                                  53              63 D0 000F6       MOVL    (SYM), SYM                          ;
                                  E0              54 E9 000F9       BLBC    SEARCH_DONE, 13$                    ; 1843
                                  01              54 D1 000FC 16$:  CMPL    SEARCH_DONE, #1                     ; 1845
                                                  08 12 000FF       BNEQ    17$                                 ;
                                  54          08  AC D0 00101       MOVL    VAR2, VAR2_DESC_ADR                 ; 1852
                                                  52 D4 00105       CLRL    VAR2_CHAN                           ; 1853
                                                  62 11 00107       BRB     22$                                 ; 1845
                                  07          1C  A3 E9 00109 17$:  BLBC    28(SYM), 18$                        ; 1858
                                  7E         00G  8F 9A 0010D       MOVZBL  #BAS$K_ILLFIEVAR, -(SP)             ;
                                  69              01 FB 00111       CALLS   #1, BAS$$STOP                       ;
                                  54              6E 9E 00114 18$:  MOVAB   VAR2_DESC, VAR2_DESC_ADR            ; 1863
                                  50          10  A3 D0 00117       MOVL    16(SYM), R0                         ; 1864
                                                  02 18 0011B       BGEQ    19$                                 ;
                                                  50 D4 0011D       CLRL    R0                                  ;
                                  6E              50 B0 0011F 19$:  MOVW    R0, VAR2_DESC                       ;
                          02  AE        010E  8F B0 00122       MOVW    #270, VAR2_DESC+2                   ; 1865
                                  52          08  A3 D0 00128       MOVL    8(SYM), VAR2_CHAN                   ; 1867
                                                  03 12 0012C       BNEQ    20$                                 ; 1869
                                  52              07 CE 0012E       MNEGL   #7, VAR2_CHAN                       ;
                                  50              07 CE 00131 20$:  MNEGL   #7, R0                              ; 1871
```

```
                                          00000000G  00  16 00134         JSB     BAS$$CB_PUSH                                 ;
                                FF4C   CB        0C  A5  D0 0013A         MOVL    12(FMP), -180(CCB)                           ; 1872
                    04   AE       EC   AB        0C  A3  C1 00140         ADDL3   12(SYM), -20(CCB), VAR2_DESC+4               ; 1873
                                       07        FC  AB  E8 00147         BLBS    -4(CCB), 21$                                 ; 1875
                                       7E       00G  8F  9A 0014B         MOVZBL  #BAS$K_IO_CHANOT, -(SP)                      ;
                                       69            01  FB 0014F         CALLS   #1, BAS$$STOP                                ;
                         53       0C   A3        10  A3  C1 00152 21$:    ADDL3   16(SYM), 12(SYM), R3                         ; 1877
           53       D2   AB            10            00  ED 00158         CMPZV   #0, #16, -46(CCB), R3                        ;
                                                     0B  1E 0015E         BGEQU   22$                                          ;
                                       7E       00G  8F  9A 00160         MOVZBL  #BAS$K_FIEOVEBUF, -(SP)                      ; 1879
                            00000000G  00            01  FB 00164         CALLS   #1, BAS$$STOP_IO                             ;
                         01            00        04  AC  CF 0016B 22$:    CASEL   STMT_TYPE, #0, #1                            ; 1888
                                     0011          0004     00170 23$:    .WORD   24$-23$,-                                    ;
                                                                                  25$-23$                                      ;
                                               0090  8F  BB 00174 24$:    PUSHR   #^M<R4,R7>                                   ; 1892
                            00000000G  00            02  FB 00178         CALLS   #2, STR$COPY_DX                              ;
                                                     0B  11 0017F         BRB     26$                                          ;
                                               0090  8F  BB 00181 25$:    PUSHR   #^M<R4,R7>                                   ; 1895
                            00000000G  00            02  FB 00185         CALLS   #2, BAS$RSET                                 ;
                                                     52  D5 0018C 26$:    TSTL    VAR2_CHAN                                    ; 1902
                                                     0C  13 0018E         BEQL    27$                                          ;
                                          00000000G  00  16 00190         JSB     BAS$$CB_GET                                  ; 1905
                                          00000000G  00  16 00196         JSB     BAS$$CB_POP                                  ; 1906
                                                     56  D5 0019C 27$:    TSTL    VAR1_CHAN                                    ; 1909
                                                     0C  13 0019E         BEQL    28$                                          ;
                                          00000000G  00  16 001A0         JSB     BAS$$CB_GET                                  ; 1912
                                          00000000G  00  16 001A6         JSB     BAS$$CB_POP                                  ; 1913
                                                         04 001AC 28$:    RET                                                  ; 1916

; Routine Size:  429 bytes,    Routine Base:  _BAS$CODE + 023D


;  893          1917  1
```

```
1 GLOBAL ROUTINE BAS$FIELD_COP_R (                         ! Copy to a FIELD variable
1           STMT_TYPE,                                     ! Either LSET or RSET
1           VAR2,                                          ! The destination variable
1           VAR1_LEN,                                      ! Length of the source variable
1           VAR1_ADDR                                      ! Address of the source variable
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This is an alternate entry point for BAS$FIELD_COPY, which the
1 !         compiler uses to avoid having to build a descriptor for a string
1 !         constant.  This code builds the descriptor and calls BAS$FIELD_COPY.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         STMT_TYPE.rl.v  0 = this is an LSET statement, 1 = RSET
1 !         VAR2.wt.dx      The destination of the copy.  This may be a
1 !                         FIELD variable.
1 !         VAR1_LEN.rl.v   The number of bytes in the source
1 !         VAR1_ADDR.rt.r  The address of the source
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         SYM$Q_ROOT.rq   The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         May write into or read from an I/O buffer.
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         VAR1 : BLOCK [8, BYTE];                          ! Build source descriptor here
2
2     VAR1 [DSC$W_LENGTH] = .VAR1_LEN;
2     VAR1 [DSC$B_DTYPE] = DSC$K_DTYPE_T;
2     VAR1 [DSC$B_CLASS] = DSC$K_CLASS_S;
2     VAR1 [DSC$A_POINTER] = .VAR1_ADDR;
2 !+
2 ! Now do the copy.
2 !-
2     BAS$FIELD_COPY (.STMT_TYPE, .VAR2, VAR1);
1     END;                                                 ! end of BAS$FIELD_COP_R
```

```
                              0000 00000        .ENTRY  BAS$FIELD_COP_R, Save nothing                  ; 1918
            5E                  08  C2 00002        SUBL2   #8, SP                                                 ;
            6E           0C     AC  B0 00005        MOVW    VAR1_LEN, VAR1                                         ; 1964
     02     AE         010E     8F  B0 00009        MOVW    #270, VAR1+2                                           ; 1965
     04     AE           10     AC  D0 0000F        MOVL    VAR1_ADDR, VAR1+4                                      ; 1967
                                5E  DD 00014        PUSHL   SP                                                     ; 1971
            7E           04     AC  7D 00016        MOVQ    STMT_TYPE, -(SP)                                       ;
   FE34     CF                  03  FB 0001A        CALLS   #3, BAS$FIELD_COPY                                     ;
                                    04 0001F        RET                                                            ; 1972

; Routine Size:  32 bytes,    Routine Base:  _BAS$CODE + 03EA


;  950        1973  1
```

```
1 GLOBAL ROUTINE BAS$FIELD_PURGE (                       ! Purge field variables
1           DECL                                         ! Scope of the declaration
1       ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Purge, or undeclare, field variables.  This routine is called
1 !         at the end of a block with declarations that might have been
1 !         FIELD variables.  It scans the symbol table and purges each
1 !         entry marked as declared in the block.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         DECL.rl.v       An indication of the scope of the declaration
1 !                         of the variable.  This is a pointer to the major
1 !                         frame (R11) if the variable is in the scope of
1 !                         the major procedure, or a pointer to the minor
1 !                         frame (R10) if the variable is in the scope of
1 !                         a DEF.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         SYM$Q_ROOT.mq
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         May remove symbols from the symbol table.
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2         SEARCH_DONE;
2
2 !+
2 ! If the symbol table root has not yet been initialized, initialize it.
2 !-
2
3     IF (.SYM$Q_ROOT [0] EQL 0)
2     THEN
3         BEGIN
3
3         LOCAL
3             AST_STATUS;
3
```

```
3              AST_STATUS = $SETAST (ENBFLG = 0);
3
4              IF (.SYM$Q_ROOT [0] EQL 0)
3              THEN
4                  BEGIN
4                  SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
3                  END;
3
3              IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
3
2              END;
2
2      !+
2      ! Search the queue, removing any variables declared in this block.
2      !-
2          SYM = .SYM$Q_ROOT [0];
2          SEARCH_DONE = 0;
2
2          DO
3              BEGIN
3
4              IF (.SYM EQLA SYM$Q_ROOT)
3              THEN
3                  SEARCH_DONE = 1
3              ELSE
3
4                  IF (.SYM [SYM$L_DECL] EQL .DECL)
3                  THEN
4                      BEGIN
4      !+
4      ! We must delete this symbol from the symbol table.
4      !-
4
4                      BUILTIN
4                          REMQUE;
4
4                      LOCAL
4                          FREE_VM_STATUS,
4                          TEMP,
4                          VAR : REF BLOCK [8, BYTE];
4
4                      REMQUE (.SYM, TEMP);
4
4                      IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
4
4                      VAR = .SYM [SYM$A_VAR];
4                      VAR [DSC$W_LENGTH] = 0;
4                      VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
4                      VAR [DSC$A_POINTER] = 0;
4                      FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
4
4                      IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
4
4                      SYM = .SYM$Q_ROOT [0];
4                      END
3                  ELSE
3                      SYM = .SYM [SYM$A_NEXT];
```

```
; 1066          2088  3
; 1067          2089  3            END
; 1068          2090  2     UNTIL (.SEARCH_DONE);
; 1069          2091  2
; 1070          2092  1     END;                                              ! end of BAS$FIELD_PURGE




                                       007C 00000          .ENTRY  BAS$FIELD_PURGE, Save R2,R3,R4,R5,R6        ; 1974
                 56 00000000G    00   9E 00002          MOVAB   BAS$$STOP, R6                               ;
                 55 00000000G    00   9E 00009          MOVAB   SYS$SETAST, R5                              ;
                 54 00000000'    EF   9E 00010          MOVAB   SYM$Q_ROOT, R4                              ;
                 5E              08   C2 00017          SUBL2   #8, SP                                      ;
                                 64   D5 0001A          TSTL    SYM$Q_ROOT                                  ; 2024
                                 1D   12 0001C          BNEQ    2$                                          ;
                                 7E   D4 0001E          CLRL    -(SP)                                       ; 2031
                 65              01   FB 00020          CALLS   #1, SYS$SETAST                              ;
                                 64   D5 00023          TSTL    SYM$Q_ROOT                                  ; 2033
                                 0A   12 00025          BNEQ    1$                                          ;
                 51              64   9E 00027          MOVAB   SYM$Q_ROOT, R1                              ; 2036
            04   A4              51   D0 0002A          MOVL    R1, SYM$Q_ROOT+4                            ;
                 64              51   D0 0002E          MOVL    R1, SYM$Q_ROOT                              ;
                 09              50   D1 00031 1$:      CMPL    AST_STATUS, #9                              ; 2039
                                 05   12 00034          BNEQ    2$                                          ;
                                 01   DD 00036          PUSHL   #1                                          ;
                 65              01   FB 00038          CALLS   #1, SYS$SETAST                              ;
                 52              64   D0 0003B 2$:      MOVL    SYM$Q_ROOT, SYM                             ; 2046
                                 53   D4 0003E          CLRL    SEARCH_DONE                                 ; 2047
                 50              64   9E 00040 3$:      MOVAB   SYM$Q_ROOT, R0                              ; 2052
                 50              52   D1 00043          CMPL    SYM, R0                                     ;
                                 05   12 00046          BNEQ    4$                                          ;
                 53              01   D0 00048          MOVL    #1, SEARCH_DONE                             ; 2054
                                 46   11 0004B          BRB     8$                                          ;
            04   AC         14   A2   D1 0004D 4$:      CMPL    20(SYM), DECL                               ; 2057
                                 3C   12 00052          BNEQ    7$                                          ;
            04   AE              62   0F 00054          REMQUE  (SYM), TEMP                                 ; 2072
                 07         1C   A2   E9 00058          BLBC    28(SYM), 5$                                 ; 2074
                 7E        00G   8F   9A 0005C          MOVZBL  #BAS$K_ILLFIEVAR, -(SP)                     ;
                 66              01   FB 00060          CALLS   #1, BAS$$STOP                               ;
                 50         18   A2   D0 00063 5$:      MOVL    24(SYM), VAR                                ; 2076
                                 60   B4 00067          CLRW    (VAR)                                       ; 2077
            03   A0              02   90 00069          MOVB    #2, 3(VAR)                                  ; 2078
                            04   A0   D4 0006D          CLRL    4(VAR)                                      ; 2079
                            04   AE   9F 00070          PUSHAB  TEMP                                        ; 2080
            04   AE              20   D0 00073          MOVL    #32, 4(SP)                                  ;
                            04   AE   9F 00077          PUSHAB  4(SP)                                       ;
     00000000G   00              02   FB 0007A          CALLS   #2, LIB$FREE_VM                             ;
                 07              50   E8 00081          BLBS    FREE_VM_STATUS, 6$                          ; 2082
                 7E        00G   8F   9A 00084          MOVZBL  #BAS$K_PROLOSSOR, -(SP)                     ;
                 66              01   FB 00088          CALLS   #1, BAS$$STOP                               ;
                 52              64   D0 0008B 6$:      MOVL    SYM$Q_ROOT, SYM                             ; 2084
                                 03   11 0008E          BRB     8$                                          ; 2057
                 52              62   D0 00090 7$:      MOVL    (SYM), SYM                                  ; 2087
                 AA              53   E9 00093 8$:      BLBC    SEARCH_DONE, 3$                             ; 2090
                                      04 00096          RET                                                 ; 2092
```

```
; Routine Size:  151 bytes,    Routine Base:  _BAS$CODE + 040A

; 1071          2093  1
```

```
1 GLOBAL ROUTINE BAS$FIELD_OPEN (                       ! Account for OPENing a file
1         CHAN                                          ! Channel just OPENed
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Account for OPENing a file.  If the record length is shorter
1 !       than before, some variables may have to be un-fielded.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       CHAN.rl.v        The channel number of the file just opened.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       SYM$Q_ROOT.mq
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May remove symbols from the symbol table.
1 !
1 !--
1
2     BEGIN
2
2     BUILTIN
2         FP;
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     LOCAL
2         FMP : REF BLOCK [, BYTE],
2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2         SEARCH_DONE,
2         LUN_NO,
2         RSZ,
2         RBF;
2
2     FMP = .FP;
2 !+
2 ! Compute the logical unit number from the channel number.
2 !-
2
2     IF (.CHAN LSS 0) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
2
2     IF (.CHAN EQL 0) THEN LUN_NO = LUB$K_LUN_INPU ELSE LUN_NO = .CHAN;
```

```
; 1130          2151  2
; 1131          2152  2  !+
; 1132          2153  2  ! If the symbol table root has not yet been initialized, initialize it.
; 1133          2154  2  !-
; 1134          2155  2
; 1135          2156  3      IF (.SYM$Q_ROOT [0] EQL 0)
; 1136          2157  2      THEN
; 1137          2158  3          BEGIN
; 1138          2159  3
; 1139          2160  3          LOCAL
; 1140          2161  3              AST_STATUS;
; 1141          2162  3
; 1142          2163  3          AST_STATUS = $SETAST (ENBFLG = 0);
; 1143          2164  3
; 1144          2165  4          IF (.SYM$Q_ROOT [0] EQL 0)
; 1145          2166  3          THEN
; 1146          2167  4              BEGIN
; 1147          2168  4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
; 1148          2169  3              END;
; 1149          2170  3
; 1150          2171  3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
; 1151          2172  3
; 1152          2173  2          END;
; 1153          2174  2
; 1154          2175  2  !+
; 1155          2176  2  ! Pick up the buffer size to compare against the variables.
; 1156          2177  2  !-
; 1157          2178  2      BAS$$CB_PUSH (.LUN_NO, LUB$K_LUN_INPU);
; 1158          2179  2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
; 1159          2180  2      RBF = .CCB [LUB$A_RBUF_ADR];
; 1160          2181  2      RSZ = .CCB [LUB$W_RBUF_SIZE];
; 1161          2182  2  !+
; 1162          2183  2  ! Search the queue, removing any variables which no longer fit in
; 1163          2184  2  ! the current buffer.
; 1164          2185  2  !-
; 1165          2186  2      SYM = .SYM$Q_ROOT [0];
; 1166          2187  2      SEARCH_DONE = 0;
; 1167          2188  2
; 1168          2189  2      DO
; 1169          2190  3          BEGIN
; 1170          2191  3
; 1171          2192  4          IF (.SYM EuLA SYM$Q_ROOT)
; 1172          2193  3          THEN
; 1173          2194  3              SEARCH_DONE = 1
; 1174          2195  3          ELSE
; 1175          2196  3
; 1176          2197  4              IF (.SYM [SYM$L_CHAN] EQL .CHAN)
; 1177          2198  3              THEN
; 1178          2199  4                  BEGIN
; 1179          2200  4
; 1180          2201  5                  IF (.SYM [SYM$L_OFFSET] + .SYM [SYM$L_LEN] LEQ .RSZ)
; 1181          2202  4                  THEN
; 1182          2203  5                      BEGIN
; 1183          2204  5  !+
; 1184          2205  5  ! The variable is still within the buffer, recompute its address,
; 1185          2206  5  ! since the buffer may have been reallocated.
; 1186          2207  5  !-
; 1187          2208  5
; 1188          2209  5                     LOCAL
; 1189          2210  5                         VAR : REF BLOCK [8, BYTE];
; 1190          2211  5
; 1191          2212  5                     VAR = .SYM [SYM$A_VAR];
; 1192          2213  5                     VAR [DSC$A_POINTER] = .RBF + .SYM [SYM$L_OFFSET];
; 1193          2214  5 !+
; 1194          2215  5 ! Clear the "invalid" bit, since it may have been set by an implied close.
; 1195          2216  5 !-
; 1196          2217  5                     SYM [SYM$V_INVALID] = 0;
; 1197          2218  5                     SYM = .SYM [SYM$A_NEXT];
; 1198          2219  5                     END
; 1199          2220  4                 ELSE
; 1200          2221  5                     BEGIN
; 1201          2222  5 !+
; 1202          2223  5 ! This variable is outside the new buffer, remove it.
; 1203          2224  5 !-
; 1204          2225  5
; 1205          2226  5                     BUILTIN
; 1206          2227  5                         REMQUE;
; 1207          2228  5
; 1208          2229  5                     LOCAL
; 1209          2230  5                         FREE_VM_STATUS,
; 1210          2231  5                         TEMP,
; 1211          2232  5                         VAR : REF BLOCK [8, BYTE];
; 1212          2233  5
; 1213          2234  5                     REMQUE (.SYM, TEMP);
; 1214          2235  5                     VAR = .SYM [SYM$A_VAR];
; 1215          2236  5                     VAR [DSC$W_LENGTH] = 0;
; 1216          2237  5                     VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
; 1217          2238  5                     VAR [DSC$A_POINTER] = 0;
; 1218          2239  5                     FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
; 1219          2240  5
; 1220          2241  5                     IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
; 1221          2242  5
; 1222          2243  5                     SYM = .SYM$Q_ROOT [0];
; 1223          2244  5                     END
; 1224          2245  5
; 1225          2246  4                 END
; 1226          2247  3             ELSE
; 1227          2248  3                 SYM = .SYM [SYM$A_NEXT];
; 1228          2249  3
; 1229          2250  3         END
; 1230          2251  2     UNTIL (.SEARCH_DONE);
; 1231          2252  2
; 1232          2253  2 !+
; 1233          2254  2 ! We are through with register CCB.
; 1234          2255  2 !-
; 1235          2256  2     BAS$$CB_POP ();
; 1236          2257  1     END;                                    ! end of BAS$FIELD_OPEN



                                             09FC 00000          .ENTRY  BAS$FIELD_OPEN, Save R2,R3,R4,R5,R6,R7,R8,-  ; 2094
                                                                         R11                                          ;
```

```
              58 00000000G  00  9E 00002         MOVAB   SYS$SETAST, R8                         ;
              57 00000000G  00  9E 00009         MOVAB   BAS$$STOP, R7                          ;
              56 00000000'  EF  9E 00010         MOVAB   SYM$Q_ROOT, R6                         ;
              5E            08  C2 00017         SUBL2   #8, SP                                 ;
              53            5D  D0 0001A         MOVL    FP, FMP                                ; 2143
              52        04  AC  D0 0001D         MOVL    CHAN, R2                               ; 2148
                            07  18 00021         BGEQ    1$                                     ;
              7E       00G  8F  9A 00023         MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                ;
              67            01  FB 00027         CALLS   #1, BAS$$STOP                          ;
                            52  D5 0002A 1$:     TSTL    R2                                     ; 2150
                            03  12 0002C         BNEQ    2$                                     ;
              52            07  CE 0002E         MNEGL   #7, LUN_NO                             ;
                            66  D5 00031 2$:     TSTL    SYM$Q_ROOT                             ; 2156
                            1D  12 00033         BNEQ    4$                                     ;
                            7E  D4 00035         CLRL    -(SP)                                  ; 2163
              68            01  FB 00037         CALLS   #1, SYS$SETAST                         ;
                            66  D5 0003A         TSTL    SYM$Q_ROOT                             ; 2165
                            0A  12 0003C         BNEQ    3$                                     ;
              51            66  9E 0003E         MOVAB   SYM$Q_ROOT, R1                         ; 2168
           04 A6            51  D0 00041         MOVL    R1, SYM$Q_ROOT+4                       ;
              66            51  D0 00045         MOVL    R1, SYM$Q_ROOT                         ;
              09            50  D1 00048 3$:     CMPL    AST_STATUS, #9                         ; 2171
                            05  12 0004B         BNEQ    4$                                     ;
                            01  DD 0004D         PUSHL   #1                                     ;
              68            01  FB 0004F         CALLS   #1, SYS$SETAST                         ;
              50            07  CE 00052 4$:     MNEGL   #7, R0                                 ; 2178
                 00000000G  00  16 00055         JSB     BAS$$CB_PUSH                           ;
         FF4C CB        0C  A3  D0 0005B         MOVL    12(FMP), -180(CCB)                     ; 2179
              54        EC  AB  D0 00061         MOVL    -20(CCB), RBF                          ; 2180
              55        D2  AB  3C 00065         MOVZWL  -46(CCB), RSZ                          ; 2181
              52            66  D0 00069         MOVL    SYM$Q_ROOT, SYM                        ; 2186
                            53  D4 0006C         CLRL    SEARCH_DONE                            ; 2187
              50            66  9E 0006E 5$:     MOVAB   SYM$Q_ROOT, R0                         ; 2192
              50            52  D1 00071         CMPL    SYM, R0                                ;
                            05  12 00074         BNEQ    6$                                     ;
              53            01  D0 00076         MOVL    #1, SEARCH_DONE                        ; 2194
                            56  11 00079         BRB     10$                                    ;
           04 AC        08  A2  D1 0007B 6$:     CMPL    8(SYM), CHAN                           ; 2197
                            4C  12 00080         BNEQ    9$                                     ;
50         0C A2        10  A2  C1 00082         ADDL3   16(SYM), 12(SYM), R0                   ; 2201
              55            50  D1 00088         CMPL    R0, RSZ                                ;
                            10  14 0008B         BGTR    7$                                     ;
              50        18  A2  D0 0008D         MOVL    24(SYM), VAR                           ; 2212
           04 A0        0C B244 9E 00091         MOVAB   @12(SYM)[RBF], 4(VAR)                  ; 2213
           1C A2            01  8A 00097         BICB2   #1, 28(SYM)                            ; 2217
                            31  11 0009B         BRB     9$                                     ; 2218
           04 AE            62  0F 0009D 7$:     REMQUE  (SYM), TEMP                            ; 2234
              50        18  A2  D0 000A1         MOVL    24(SYM), VAR                           ; 2235
                            60  B4 000A5         CLRW    (VAR)                                  ; 2236
           03 A0            02  90 000A7         MOVB    #2, 3(VAR)                             ; 2237
                        04  A0  D4 000AB         CLRL    4(VAR)                                 ; 2238
                        04  AE  9F 000AE         PUSHAB  TEMP                                   ; 2239
           04 AE            20  D0 000B1         MOVL    #32, 4(SP)                             ;
                        04  AE  9F 000B5         PUSHAB  4(SP)                                  ;
    00000000G 00            02  FB 000B8         CALLS   #2, LIB$FREE_VM                        ;
              07            50  E8 000BF         BLBS    FREE_VM_STATUS, 8$                     ; 2241
              7E       00G  8F  9A 000C2         MOVZBL  #BAS$K_PROLOSSOR, -(SP)                ;
```

```
                       67          01  FB 000C6          CALLS   #1, BAS$$STOP                              ;
                       52          66  D0 000C9 8$:      MOVL    SYM$Q_ROOT, SYM                            ; 2243
                                   03  11 000CC          BRB     10$                                        ; 2199
                       52          62  D0 000CE 9$:      MOVL    (SYM), SYM                                 ; 2248
                       9A          53  E9 000D1 10$:     BLBC    SEARCH_DONE, 5$                            ; 2251
                        00000000G  00  16 000D4          JSB     BAS$$CB_POP                                ; 2256
                                       04 000DA          RET                                                ; 2257

; Routine Size:  219 bytes,    Routine Base:  _BAS$CODE + 04A1


; 1237          2258  1
```

```
1 GLOBAL ROUTINE BAS$FIELD_CLOSE (                          ! Account for CLOSEing a file
1         CHAN                                              ! Channel about to be CLOSEed
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Account for CLOSEing a file.  Unfield all of the variables
1 !       on this channel.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       CHAN.rl.v       The channel number of the file about to be
1 !                       CLOSEed.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       SYM$Q_ROOT.mq
1 !       LUB$V_FIELD_USE for this channel, set to 0
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May remove symbols from the symbol table.
1 !
1 !--
1
2     BEGIN
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     LOCAL
2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2         LUN_NO,
2         SEARCH_DONE;
2
2 !+
2 ! If the symbol table root has not yet been initialized, initialize it.
2 !-
2
3     IF (.SYM$Q_ROOT [0] EQL 0)
2     THEN
3         BEGIN
3
3         LOCAL
3             AST_STATUS;
3
3         AST_STATUS = $SETAST (ENBFLG = 0);
```

```
3
4          IF (.SYM$Q_ROOT [0] EQL 0)
3          THEN
4              BEGIN
4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
3              END;
3
3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
3
2          END;
2
2  !+
2  ! Search the queue, removing any variables for this channel.
2  !-
2      SYM = .SYM$Q_ROOT [0];
2      SEARCH_DONE = 0;
2
2      DO
3          BEGIN
3
4          IF (.SYM EQLA SYM$Q_ROOT)
3          THEN
3              SEARCH_DONE = 1
3          ELSE
3
4              IF (.SYM [SYM$L_CHAN] EQL .CHAN)
3              THEN
4                  BEGIN
4  !+
4  ! We must delete this symbol from the symbol table.
4  !-
4
4                  BUILTIN
4                      REMQUE;
4
4                  LOCAL
4                      FREE_VM_STATUS,
4                      TEMP,
4                      VAR : REF BLOCK [8, BYTE];
4
4                  REMQUE (.SYM, TEMP);
4
4                  IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
4
4                  VAR = .SYM [SYM$A_VAR];
4                  VAR [DSC$W_LENGTH] = 0;
4                  VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
4                  VAR [DSC$A_POINTER] = 0;
4                  FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
4
4                  IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
4
4                  SYM = .SYM$Q_ROOT [0];
4                  END
3              ELSE
3                  SYM = .SYM [SYM$A_NEXT];
3
```

```
; 1353          2373  3             END
; 1354          2374  2         UNTIL (.SEARCH_DONE);
; 1355          2375  2
; 1356          2376  2     !+
; 1357          2377  2     ! load register CCB.
; 1358          2378  2     !-
; 1359          2379  2         LUN_NO = (IF (.CHAN EQL 0) THEN LUB$K_LUN_INPU ELSE .CHAN);
; 1360          2380  2         BAS$$CB_PUSH (.LUN_NO, LUB$K_ILUN_MIN);
; 1361          2381  2
; 1362          2382  2     !+
; 1363          2383  2     ! indicate there is not FIELDing on this channel anymore.
; 1364          2384  2     !-
; 1365          2385  2         CCB [LUB$V_FIELD_USE] = 0;
; 1366          2386  2
; 1367          2387  2     !+
; 1368          2388  2     ! done with register CCB.
; 1369          2389  2     !-
; 1370          2390  2         BAS$$CB_POP ();
; 1371          2391  2
; 1372          2392  1         END;                                            ! end of BAS$FIELD_CLOSE
```

```
                                    083C 00000            .ENTRY   BAS$FIELD_CLOSE, Save R2,R3,R4,R5,R11          ; 2259
              55 00000000G    00    9E 00002            MOVAB    BAS$$STOP, R5                                  ;
              54 00000000G    00    9E 00009            MOVAB    SYS$SETAST, R4                                 ;
              53 00000000'    EF    9E 00010            MOVAB    SYM$Q_ROOT, R3                                 ;
              5E              08    C2 00017            SUBL2    #8, SP                                         ;
                              63    D5 0001A            TSTL     SYM$Q_ROOT                                     ; 2308
                              1D    12 0001C            BNEQ     2$                                             ;
                              7E    D4 0001E            CLRL     -(SP)                                          ; 2315
              64              01    FB 00020            CALLS    #1, SYS$SETAST                                 ;
                              63    D5 00023            TSTL     SYM$Q_ROOT                                     ; 2317
                              0A    12 00025            BNEQ     1$                                             ;
              51              63    9E 00027            MOVAB    SYM$Q_ROOT, R1                                 ; 2320
          04  A3              51    D0 0002A            MOVL     R1, SYM$Q_ROOT+4                               ;
              63              51    D0 0002E            MOVL     R1, SYM$Q_ROOT                                 ;
              09              50    D1 00031 1$:        CMPL     AST_STATUS, #9                                 ; 2323
                              05    12 00034            BNEQ     2$                                             ;
                              01    DD 00036            PUSHL    #1                                             ;
              64              01    FB 00038            CALLS    #1, SYS$SETAST                                 ;
              52              63    D0 0003B 2$:        MOVL     SYM$Q_ROOT, SYM                                ; 2330
                              5B    D4 0003E            CLRL     SEARCH_DONE                                    ; 2331
              50              63    9E 00040 3$:        MOVAB    SYM$Q_ROOT, R0                                 ; 2336
              50              52    D1 00043            CMPL     SYM, R0                                        ;
                              05    12 00046            BNEQ     4$                                             ;
              5B              01    D0 00048            MOVL     #1, SEARCH_DONE                                ; 2338
                              46    11 0004B            BRB      8$                                             ;
          04  AC        08    A2    D1 0004D 4$:        CMPL     8(SYM), CHAN                                   ; 2341
                              3C    12 00052            BNEQ     7$                                             ;
          04  AE              62    0F 00054            REMQUE   (SYM), TEMP                                    ; 2356
              07        1C    A2    E9 00058            BLBC     28(SYM), 5$                                    ; 2358
              7E       00G    8F    9A 0005C            MOVZBL   #BAS$K_ILLFIEVAR, -(SP)                        ;
              65              01    FB 00060            CALLS    #1, BAS$$STOP                                  ;
              50        18    A2    D0 00063 5$:        MOVL     24(SYM), VAR                                   ; 2360
```

```
                               60  B4 00067          CLRW    (VAR)                          ; 2361
       03  A0                  02  90 00069          MOVB    #2, 3(VAR)                     ; 2362
                     04        A0  D4 0006D          CLRL    4(VAR)                         ; 2363
                     04        AE  9F 00070          PUSHAB  TEMP                           ; 2364
       04  AE                  20  D0 00073          MOVL    #32, 4(SP)                     ;
                     04        AE  9F 00077          PUSHAB  4(SP)                          ;
00000000G  00                  02  FB 0007A          CALLS   #2, LIB$FREE_VM                ;
           07                  50  E8 00081          BLBS    FREE_VM_STATUS, 6$             ; 2366
           7E       00G        8F  9A 00084          MOVZBL  #BAS$K_PROLOSSOR, -(SP)        ;
           65                  01  FB 00088          CALLS   #1, BAS$$STOP                  ;
           52                  63  D0 0008B 6$:      MOVL    SYM$Q_ROOT, SYM                ; 2368
                               03  11 0008E          BRB     8$                             ; 2341
           52                  62  D0 00090 7$:      MOVL    (SYM), SYM                     ; 2371
           AA                  5B  E9 00093 8$:      BLBC    SEARCH_DONE, 3$                ; 2374
                     04        AC  D5 00096          TSTL    CHAN                           ; 2379
                               05  12 00099          BNEQ    9$                             ;
           52                  07  CE 0009B          MNEGL   #7, LUN_NO                     ;
                               04  11 0009E          BRB     10$                            ;
           52        04        AC  D0 000A0 9$:      MOVL    CHAN, LUN_NO                   ;
           50                  08  CE 000A4 10$:     MNEGL   #8, R0                         ; 2380
              00000000G        00  16 000A7          JSB     BAS$$CB_PUSH                   ;
       A1  AB        40        8F  8A 000AD          BICB2   #64, -95(CCB)                  ; 2385
              00000000G        00  16 000B2          JSB     BAS$$CB_POP                    ; 2390
                                   04 000B8          RET                                    ; 2392

; Routine Size:  185 bytes,    Routine Base:  _BAS$CODE + 057C


; 1373         2393  1
```

```
1 ROUTINE BAS$$FIELD_KILL                                        ! CLOSE appendage
1     : CALL_CCB NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called while a file is being CLOSEd, for any
1 !       reason.  If the CLOSE was explicit and in the module containing
1 !       the FIELD statement(s), BAS$FIELD_CLOSE will already have
1 !       removed all of the field variables for this channel from the
1 !       symbol table, so this routine will find none.  If the CLOSE
1 !       is implicit or outside the module with the FIELD statement(s),
1 !       BAS$FIELD_CLOSE will not have been called and this routine
1 !       will mark some variables invalid.  An explicit CLOSE from
1 !       another module is considered a programming error, so it is
1 !       proper to give an error as soon as any of these variables are
1 !       referenced.  We cannot signal an error from here because this
1 !       may be the CLOSE from the exit handler (in which case the
1 !       variables will not be referenced again, so marking them
1 !       invalid is OK) or the implicit CLOSE from OPEN, in which case
1 !       (if the OPEN is from a module with FIELD) BAS$FIELD_OPEN will
1 !       re-validate the variables still in the buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !       LUB$W_LUN       The logical unit number of the file being closed
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       SYM$Q_ROOT.mq
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May mark symbols invalid, but is most likely to have no net
1 !       effect.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         CCB : REF BLOCK [, BYTE];
2
2     LOCAL
2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2         SEARCH_DONE,
2         CHAN;
```

```
2
2   !+
2   ! If the symbol table root has not yet been initialized, initialize it.
2   !-
2
3       IF (.SYM$Q_ROOT [0] EQL 0)
2       THEN
3           BEGIN
3
3           LOCAL
3               AST_STATUS;
3
3           AST_STATUS = $SETAST (ENBFLG = 0);
3
4           IF (.SYM$Q_ROOT [0] EQL 0)
3           THEN
4               BEGIN
4               SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
3               END;
3
3           IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
3
2           END;
2
2   !+
2   ! Compute the channel number from the logical unit number.
2   !-
2       CHAN = (IF (.CCB [LUB$W_LUN] EQL LUB$K_LUN_INPU) THEN 0 ELSE .CCB [LUB$W_LUN]);
2   !+
2   ! Search the queue, invalidating any variables for this channel.
2   !-
2       SYM = .SYM$Q_ROOT [0];
2       SEARCH_DONE = 0;
2
2       DO
3           BEGIN
3
4           IF (.SYM EQLA SYM$Q_ROOT)
3           THEN
3               SEARCH_DONE = 1
3           ELSE
3
4               IF (.SYM [SYM$L_CHAN] EQL .CHAN)
3               THEN
4                   BEGIN
4   !+
4   ! We must mark this symbol as invalid.
4   !-
4
4                   LOCAL
4                       VAR : REF BLOCK [8, BYTE];
4
4                   VAR = .SYM [SYM$A_VAR];
4                   VAR [DSC$A_POINTER] = 0;
4                   SYM [SYM$V_INVALID] = 1;
3                   END;
3
```

```
; 1489          2508  3         SYM = .SYM [SYM$A_NEXT];
; 1490          2509  3         END
; 1491          2510  2     UNTIL (.SEARCH_DONE);
; 1492          2511  2
; 1493          2512  1     END;                                        ! end of BAS$$FIELD_KILL



                                                       003C 00000 BAS$$FIELD_KILL:
                                                                          .WORD   Save R2,R3,R4,R5      ; 2394
                                       55 00000000G  00  9E 00002         MOVAB   SYS$SETAST, R5        ;
                                       54 00000000'  EF  9E 00009         MOVAB   SYM$Q_ROOT, R4        ;
                                                     64  D5 00010         TSTL    SYM$Q_ROOT            ; 2456
                                                     1D  12 00012         BNEQ    2$                    ;
                                                     7E  D4 00014         CLRL    -(SP)                 ; 2463
                                       65            01  FB 00016         CALLS   #1, SYS$SETAST        ;
                                                     64  D5 00019         TSTL    SYM$Q_ROOT            ; 2465
                                                     0A  12 0001B         BNEQ    1$                    ;
                                       51            64  9E 0001D         MOVAB   SYM$Q_ROOT, R1        ; 2468
                                  04   A4            51  D0 00020         MOVL    R1, SYM$Q_ROOT+4      ;
                                       64            51  D0 00024         MOVL    R1, SYM$Q_ROOT        ;
                                       09            50  D1 00027 1$:     CMPL    AST_STATUS, #9        ; 2471
                                                     05  12 0002A         BNEQ    2$                    ;
                                                     01  DD 0002C         PUSHL   #1                    ;
                                       65            01  FB 0002E         CALLS   #1, SYS$SETAST        ;
                                FFFF9  8F        C6  AB  B1 00031 2$:     CMPW    -58(CCB), #-7         ; 2478
                                                     04  12 00037         BNEQ    3$                    ;
                                                     52  D4 00039         CLRL    CHAN                  ;
                                                     04  11 0003B         BRB     4$                    ;
                                       52        C6  AB  32 0003D 3$:     CVTWL   -58(CCB), CHAN        ;
                                       51            64  D0 00041 4$:     MOVL    SYM$Q_ROOT, SYM       ; 2482
                                                     53  D4 00044         CLRL    SEARCH_DONE           ; 2483
                                       50            64  9E 00046 5$:     MOVAB   SYM$Q_ROOT, R0        ; 2488
                                       50            51  D1 00049         CMPL    SYM, R0               ;
                                                     05  12 0004C         BNEQ    6$                    ;
                                       53            01  D0 0004E         MOVL    #1, SEARCH_DONE       ; 2490
                                                     11  11 00051         BRB     7$                    ;
                                       52        08  A1  D1 00053 6$:     CMPL    8(SYM), CHAN          ; 2493
                                                     0B  12 00057         BNEQ    7$                    ;
                                       50        18  A1  D0 00059         MOVL    24(SYM), VAR          ; 2503
                                                 04  A0  D4 0005D         CLRL    4(VAR)                ; 2504
                                  1C   A1            01  88 00060         BISB2   #1, 28(SYM)           ; 2505
                                       51            61  D0 00064 7$:     MOVL    (SYM), SYM            ; 2508
                                       DC            53  E9 00067         BLBC    SEARCH_DONE, 5$       ; 2510
                                                         04 0006A         RET                           ; 2512

; Routine Size:  107 bytes,    Routine Base:  _BAS$CODE + 0635


; 1494          2513  1
```

```
1 GLOBAL ROUTINE BAS$$FIELD_INIT : NOVALUE =        ! Initialize for RUN
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Initialize the FIELD symbol table for the RUN command.  All symbols are removed
1 !       from the table, even those marked invalid.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       SYM$Q_ROOT.mq
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Makes the symbol table empty.
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
2         SEARCH_DONE;
2
2 !+
2 ! If the symbol table root has not yet been initialized, initialize it.
2 !-
2
3     IF (.SYM$Q_ROOT [0] EQL 0)
2     THEN
3         BEGIN
3
3         LOCAL
3             AST_STATUS;
3
3         AST_STATUS = $SETAST (ENBFLG = 0);
3
4         IF (.SYM$Q_ROOT [0] EQL 0)
3         THEN
4             BEGIN
4             SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
3             END;
3
3         IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
```

```
; 1553          2571  3
; 1554          2572  2         END;
; 1555          2573  2
; 1556          2574  2  !+
; 1557          2575  2  ! Search the queue, deleting any symbols in it.
; 1558          2576  2  !-
; 1559          2577  2      SYM = .SYM$Q_ROOT [0];
; 1560          2578  2      SEARCH_DONE = 0;
; 1561          2579  2
; 1562          2580  2      DO
; 1563          2581  3         BEGIN
; 1564          2582  3
; 1565          2583  4         IF (.SYM EQLA SYM$Q_ROOT)
; 1566          2584  3         THEN
; 1567          2585  3             SEARCH_DONE = 1
; 1568          2586  3         ELSE
; 1569          2587  4             BEGIN
; 1570          2588  4  !+
; 1571          2589  4  ! We must delete this symbol from the symbol table.
; 1572          2590  4  !-
; 1573          2591  4
; 1574          2592  4             BUILTIN
; 1575          2593  4                  REMQUE;
; 1576          2594  4
; 1577          2595  4             LOCAL
; 1578          2596  4                  FREE_VM_STATUS,
; 1579          2597  4                  TEMP,
; 1580          2598  4                  VAR : REF BLOCK [8, BYTE];
; 1581          2599  4
; 1582          2600  4             REMQUE (.SYM, TEMP);
; 1583          2601  4             VAR = .SYM [SYM$A_VAR];
; 1584          2602  4             FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
; 1585          2603  4
; 1586          2604  4             IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
; 1587          2605  4
; 1588          2606  4             SYM = .SYM$Q_ROOT [0];
; 1589          2607  4             END
; 1590          2608  4
; 1591          2609  3         END
; 1592          2610  2      UNTIL (.SEARCH_DONE);
; 1593          2611  2
; 1594          2612  1      END;                                                     ! end of BAS$$FIELD_INIT



                                       003C 00000         .ENTRY  BAS$$FIELD_INIT, Save R2,R3,R4,R5                  ; 2514
                         55 00000000G  00   9E 00002         MOVAB   SYS$SETAST, R5                                  ;
                         54 00000000'  EF   9E 00009         MOVAB   SYM$Q_ROOT, R4                                  ;
                         5E            08   C2 00010         SUBL2   #8, SP                                          ;
                                       64   D5 00013         TSTL    SYM$Q_ROOT                                      ; 2555
                                       1D   12 00015         BNEQ    2$                                              ;
                                       7E   D4 00017         CLRL    -(SP)                                           ; 2562
                         65            01   FB 00019         CALLS   #1, SYS$SETAST                                  ;
                                       64   D5 0001C         TSTL    SYM$Q_ROOT                                      ; 2564
                                       0A   12 0001E         BNEQ    1$                                              ;
```

```
                 51          64  9E 00020       MOVAB   SYM$Q_ROOT, R1                         ; 2567
           04    A4          51  D0 00023       MOVL    R1, SYM$Q_ROOT+4                       ;
                 64          51  D0 00027       MOVL    R1, SYM$Q_ROOT                         ;
                 09          50  D1 0002A 1$:   CMPL    AST_STATUS, #9                         ; 2570
                             05  12 0002D       BNEQ    2$                                     ;
                             01  DD 0002F       PUSHL   #1                                     ;
                 65          01  FB 00031       CALLS   #1, SYS$SETAST                         ;
                 52          64  D0 00034 2$:   MOVL    SYM$Q_ROOT, SYM                        ; 2577
                             53  D4 00037       CLRL    SEARCH_DONE                            ; 2578
                 50          64  9E 00039 3$:   MOVAB   SYM$Q_ROOT, R0                         ; 2583
                 50          52  D1 0003C       CMPL    SYM, R0                                ;
                             05  12 0003F       BNEQ    4$                                     ;
                 53          01  D0 00041       MOVL    #1, SEARCH_DONE                        ; 2585
                             2A  11 00044       BRB     6$                                     ;
           04    AE          62  0F 00046 4$:   REMQUE  (SYM), TEMP                            ; 2600
                 50     18   A2  D0 0004A       MOVL    24(SYM), VAR                           ; 2601
                        04   AE  9F 0004E       PUSHAB  TEMP                                   ; 2602
           04    AE          20  D0 00051       MOVL    #32, 4(SP)                             ;
                        04   AE  9F 00055       PUSHAB  4(SP)                                  ;
    00000000G    00          02  FB 00058       CALLS   #2, LIB$FREE_VM                        ;
                 0B          50  E8 0005F       BLBS    FREE_VM_STATUS, 5$                     ; 2604
                 7E    00G   8F  9A 00062       MOVZBL  #BAS$K_PROLOSSOR, -(SP)                ;
    00000000G    00          01  FB 00066       CALLS   #1, BAS$$STOP                          ;
                 52          64  D0 0006D 5$:   MOVL    SYM$Q_ROOT, SYM                        ; 2606
                 C6          53  E9 00070 6$:   BLBC    SEARCH_DONE, 3$                        ; 2610
                                 04 00073       RET                                            ; 2612

; Routine Size:  116 bytes,    Routine Base:  _BAS$CODE + 06A0


; 1595          2613  1
; 1596          2614  1 END                                     ! end of module BAS$RSTS_FIELD
; 1597          2615  1
; 1598          2616  0 ELUDOM




;                                 PSECT SUMMARY
;
;       Name                   Bytes                          Attributes
;
;   _BAS$DATA                      8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   _BAS$CODE                   1812  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)




;                            Library Statistics
;
;                                        -------- Symbols --------      Pages      Processing
;       File                             Total    Loaded   Percent      Mapped     Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776       12         0        581        00:01.2
;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASRSTSFI/OBJ=OBJ$:BASRSTSFI MSRC$:BASRSTSFI/UPDATE=(ENH$:BASRSTSFI
;       )

; Size:          1812 code + 8 data bytes
; Run Time:         00:40.2
; Elapsed Time:     01:22.3
; Lines/CPU Min:     3900
; Lexemes/CPU-Min: 25164
; Memory Used:  227 pages
; Compilation Complete
```

0031 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASRTDIM
LIS

BASSARITH
LIS

BASSCALE
LIS

BASSIGNAL
LIS

BASRUNINI
LIS

BASSCRATC
LIS

BASRSTSFI
LIS

BASSLEEP
LIS

BASSTOP
LIS

BASSEG
LIS